

NASA  
Contractor Report 4224

AVSCOM  
Technical Report 89-C-002

# Topology of Modified Helical Gears and Tooth Contact Analysis (TCA) Program

Faydor L. Litvin and Jiao Zhang  
*University of Illinois at Chicago*  
*Chicago, Illinois*

Prepared for  
Propulsion Directorate  
USAARTA-AVSCOM and  
NASA Lewis Research Center  
under Grant NAG3-655



National Aeronautics and  
Space Administration  
Office of Management  
Scientific and Technical  
Information Division

1989

PAGE \_\_\_\_\_ INTERNALLY BLANK

## **Abstract**

The contents of this report covers: (i) development of optimal geometry for crowned helical gears; (ii) method for their generation; (iii) tooth contact analysis (TCA) computer programs for the analysis of meshing and bearing contact of the crowned helical gears and (iv) modelling and simulation of gear shaft deflection.

The developed method for synthesis is used for determination of optimal geometry for crowned helical pinion surface and is directed to localize the bearing contact and guarantee the favorable shape and low level of the transmission errors.

Two new methods for generation of the crowned helical pinion surface have been proposed. One is based on application of the tool with a surface of revolution that slightly deviates from a regular cone surface. The tool can be used as a grinding wheel or as a shaver. Other is based on crowning pinion tooth surface with predesigned transmission errors. The pinion tooth surface can be generated by a computer controlled automatic grinding machine.

The TCA program simulates the meshing and bearing contact of the misaligned gears. The transmission errors are also determined.

The gear shaft deformation has been modelled and investigated. It has been found the deflection of gear shafts has the same effects as those of gear misalignment.

**PRECEDING PAGE BLANK NOT FILMED**

## Table of Contents

	Page
Abstract . . . . .	iii
List of Symbols . . . . .	vi
Chapter	
Summary . . . . .	1
1. Introduction . . . . .	1
2. Basic Concepts and Considerations . . . . .	2
2.1 Simulation of Meshing . . . . .	2
2.2 Simulation of Contact . . . . .	5
2.3 Partial Compensation of Transmission Errors . . . . .	6
2.4 Misalignment of Regular Helical Gears . . . . .	8
2.5 Surface Deviation by the Change of Pinion Lead . . . . .	11
3. Generation of Pinion Tooth Surface by a Surface of Revolution . . . . .	12
3.1 Basic Consideration . . . . .	12
3.2 Principle of Generation and Used Coordinate Systems . . . . .	13
3.3 Tool Surface . . . . .	15
3.4 Pinion Tooth Surface . . . . .	18
3.5 Condition of Pinion Non-Undercutting . . . . .	21
3.6 Principal Directions and Curvatures of Tooth Surface . . . . .	25
3.7 Contact Ellipse and Bearing Contact . . . . .	29
3.8 Simulation of Meshing and Determination of Transmission Errors . . . . .	30
3.9 Modification of Generating Surface $\Sigma_p$ . . . . .	35
4. Crowned Helical Pinion with Longitudinal Path Contact . . . . .	38
4.1 Basic Concepts and Considerations . . . . .	38
4.2 Method 1 . . . . .	39
4.3 Method 2 . . . . .	44
4.4 Discussion and Example . . . . .	50
5. Deformation of Helical Gear Shaft . . . . .	52
5.1 Basic Concepts and Considerations . . . . .	52
5.2 Force Applied on Gear Shaft . . . . .	53
5.3 Modelling of Shaft Deformation . . . . .	54

5.4 Example and Discussion . . . . .	59
6. Conclusion . . . . .	60
References . . . . .	62
Figures . . . . .	63
Appendix - Flowcharts and Programs . . . . .	94

## List of Symbols

$a^*$	half the length of major axis of contact ellipse
$a$	coefficient of parabolic function
$b$	addendum of cutting tool or coefficient of linear function
$b^*$	half the length of minor axis of contact ellipse
$[b_{ij}]$	3x3 auxiliary matrix
$C$	Operating center distance
$\tilde{C}$	operating center distance vector
$C^0 = \frac{N_1}{2P} + \frac{N_2}{2P}$	nominal center distance
$\tilde{C}^0$	nominal center distance vector
$d$	height of generating cone or level of transmission error.
$e_I^{(i)}, e_{II}^{(i)}$	unit vectors along principal direction of surface $\Sigma_i$ if principal directions of two surface coincide, $i$ is neglected.
$f_i(X_1, X_2, \dots)$	function expression with respect to variable $X_1, X_2, \dots$
$F_1(X_1, X_2, \dots)$	function expression with respect to variable $X_1, X_2$
$F$	magnitude of $\tilde{F}$
$\tilde{F}$	acting force between gear tooth surfaces
$\tilde{F}_t$	acting force in transverse section
$\tilde{F}_z$	acting force in axial direction
$g(\phi_1)$	auxiliary function of $\phi_1$
$i_t, j_t, k_t$	unit direction vectors of coordinate system $t$
$L_{ij}$	line of tangency of surfaces $\Sigma_i$ and $\Sigma_j$
$[L_{ij}]$	projection transformation matrix (from $S_j$ to $S_i$ )
$m_{lp} = \frac{d\phi_p}{ds}$	kinematic ratio
$m'_{lp} = \frac{dm_{lp}}{ds}$	derivative of kinematic ratio of gear and rack cutter

$m_{21} = \frac{d\phi_2}{d\phi_1} = \frac{\omega_2}{\omega_1}$	kinematic ratio of gears
$m'_{21} = \frac{dm_{21}}{d\phi_1}$	derivative of kinematic ratio of gears
$[M_{ij}]$	coordinate transformation matrix (from $S_j$ to $S_i$ )
$\dot{\tilde{n}}_r$	velocity of end point of unit normal corresponding a point moving over a surface
$\tilde{n}_i^{(j)}$	unit normal vector of surface $\Sigma_j$ in coordinate system $S_i$ [sometimes (i) is omitted if it is unnecessary to specify coordinate system]
$N_1$	number of pinion teeth
$N_2$	number of gear teeth
$O_i$	origin of coordinate system i
$P_n$	diametral pitch
$r_1$	radius of pitch circle for pinion
$r_2$	radius of pitch circle for gear
$R$	radius of arc of generating surface
$r_i^{(j)}(u, \theta)$	position vector describing surface $\Sigma_j$ with surface coordinate $(u, \theta)$ in coordinate system $S_i$ [sometimes (j) is omitted]
$S$	parameter of cutting motion
$S_i(X_i, Y_i, Z_i)$	coordinate system i
$t_p, t_G$	surface coordinates of pinion and gear
$u_i$	generating surface coordinate
$u_p^*$	auxiliary variable
$\tilde{v}_r$	velocity of a point moving over a surface
$\tilde{v}_j^{(i)}$	velocity of surface $\Sigma_i$ in coordinate system j (j is omitted sometimes)
$\tilde{v}_f^{(ij)} = \tilde{v}_f^{(i)} - \tilde{v}_f^{(j)}$	relative velocity between surface $\Sigma_i$ and $\Sigma_j$ at contact point in coordinate system f (f is omitted sometimes).
$v_I^{(ij)}, v_{II}^{(ij)}$	projection of $v^{(ij)}$ in the principal direction of surface
$x_i^{(j)}, y_i^{(j)}, z_i^{(j)}$	coordinates of vector in $S_i$ [sometimes (j) is omitted]

$\alpha$	half of cone angle or coordinate of generating surface
$\beta$	generating surface coordinate
$\beta_p$	helical angle
$\Delta\gamma$	crossing angle between axes of pinion and gear
$\Delta\delta$	intersecting angle between axes of pinion and gear
$\Delta\phi_2$	kinematic error (transmission error)
$\Delta\phi_2'$	derivative of kinematic error with respect to $\phi_1$
$\varepsilon$	elastic deformation of the contacting point
$\theta_i$	generating surface coordinate
$\kappa_I^{(i)}, \kappa_{II}^{(i)}$	principal curvatures of surface $\Sigma_i$
$\kappa_\varepsilon^{(i)} = \kappa_I^{(i)} + \kappa_{II}^{(i)}$	auxiliary function
$\lambda_i$	rotation of certain section of gear shaft
$\rho$	radius of genatrix arc for revolute surface
$\sigma^{(ij)}$	angle between $e_I^{(i)}$ and $e_I^{(j)}$
$\Sigma_i$	pinion tooth surface
$\Sigma_2$	gear tooth surface (or shape)
$\Sigma_G$	gear generating surface
$\Sigma_p$	pinion generating surface
$\phi_p$	angle of rotation for pinion being generated
$\phi_G$	angle of rotation for gear being generated
$\phi_1$	angle of pinion rotation in meshing with gear
$\phi_2$	angle of gear rotation in meshing with pinion
$\phi_{20}$	theoretical angle of gear rotation in meshing with pinion
$\psi_1, \psi_2$	new coordinates to describe parabolic function
$\psi_c, \psi_n$	pressure angles in transverse section and normal section
$\omega_1$	angular velocity of pinion
$\omega_2$	angular velocity of gear
$\omega_p$	angular velocity of pinion being generated
$\omega_G$	angular velocity of gear being generated

$$\omega^{(12)} = \omega^{(1)} - \omega^{(2)}$$

$\mu$

$v_i$

relative angular velocity

angle for installment of pinion cutting tool

deflection of certain point of gear shaft



## SUMMARY

The topology of several types of modified surfaces for helical gears are proposed. The modified surfaces allow absorption of linear or almost linear function of transmission errors caused by gear misalignment and deflection of shaft. These surfaces result in the improved contact of gear tooth surfaces. The principles and corresponding programs for computer aided simulation of meshing and contact of gears have been developed. The results of this investigation are illustrated with numerical examples.

### 1. INTRODUCTION

Traditional methods for generation of involute helical gears with parallel axes provide developed ruled tooth surfaces for the gear teeth (Fig. 1.1). The tooth surfaces contact each other at every instant along a line,  $L$ , that is the tangent to the helix on the base cylinder. The surface normals along  $L$  do not change their orientation. The disadvantage of regular helical gears is that they are very sensitive to misalignments such as the crossing or intersection of gear axes. The misaligned gears transform rotation with a linear function of transmission errors (a main source of noise) and the bearing contact is shifted to the edge of the teeth. The frequency of transmission errors coincides with the frequency of tooth meshing. The actual contact ratio (the average number of teeth being in mesh at every instant) is close to one and is far from the expected value.

These are the reasons why we have to reconsider the canonical ideas on involute helical gears and modify their tooth surfaces. Crowning of the gear surfaces is needed to negate the effects of transmission errors and the shift of contact between the gear tooth surfaces. Deviations of screw involute gear tooth surfaces to provide a new topology that can reduce the gear sensitivity to misalignment will be developed. Theoretically, the modified tooth surfaces will be in contact at every instant at a point instead of a line. Actually, due to the load applied between meshing teeth, the contact will be spread over an elliptical area whose dimensions may be controlled. Methods for gear tooth surface generation that provide the desirable surface deviation are proposed. For economical reasons only the pinion tooth surface is modified while the gear surface is kept as a regular screw involute surface.

## **2. BASIC CONCEPTS AND CONSIDERATIONS**

### **2.1 Simulation of Meshing**

The investigation of influence of gear misalignment requires a numerical solution for the simulation of meshing and contact of gear tooth surfaces. The basic ideas of this method (Litvin, 1968) are as follows:

(1) The meshing of gear tooth surfaces is considered in a fixed coordinate system,  $S_f$ . Usually, the generated gear tooth surfaces may be represented in a three parametric form with an additional relation between these parameters - Gaussian

coordinates. Such a parametric form is the result of representation of a gear tooth surface as an envelope of the family of the tool surface (the generating surface) and two from the three Gaussian coordinates are inherited from the tool surface.

The continuous tangency of gear tooth surfaces is represented by the following equations

$$\underline{r}^{(1)}(u_1, \theta_1, \psi_1, \phi_1) = \underline{r}^{(2)}(u_2, \theta_2, \psi_2, \phi_2) \quad (2.1.1)$$

$$\underline{n}^{(1)}(u_1, \theta_1, \psi_1, \phi_1) = \underline{n}^{(2)}(u_2, \theta_2, \psi_2, \phi_2), \quad |\underline{n}^{(1)}| = |\underline{n}^{(2)}| \quad (2.1.2)$$

$$f_6(u_1, \theta_1, \psi_1) = 0 \quad (2.1.3)$$

$$f_7(u_2, \theta_2, \psi_2) = 0 \quad (2.1.4)$$

Here:  $u_i$  and  $\theta_i$  are the tool surface curvilinear coordinates,  $\psi_i$  is the parameter of motion in the process of generation of the gear tooth surface,  $\phi_i$  is the angle of rotation of the gear being in mesh with the mating gear.

Equations (2.1.1) to (2.1.4) provide that the position vectors  $\underline{r}^{(1)}$  and  $\underline{r}^{(2)}$  and surface unit normals  $\underline{n}^{(1)}$  and  $\underline{n}^{(2)}$  are equal for the gear tooth surfaces in contact (Fig. 2.1). Vector equations (2.1.1) and (2.1.2) yield five independent equations and the total equation system is

$$f_i(u_1, \theta_1, \phi_1, u_2, \theta_2, \phi_2, \psi_1, \psi_2) = 0, \text{ i.e. } [1, 5],$$

$$f_6(u_1, \theta_1, \psi_1) = 0, f_7(u_2, \theta_2, \psi_2) = 0 \quad (2.1.5)$$

An instantaneous point contact instead of a line contact is guaranteed if the Jacobian differs from zero, i.e. if

$$\frac{D(f_1, f_2, f_3, f_4, f_5, f_6, f_7)}{D(u_1, \theta_1, \psi_1, u_2, \theta_2, \psi_2, \phi_2)} \neq 0 \quad (2.1.6)$$

If the inequality equation (2.1.6) is observed, then the system of equations (2.1.5) may be solved in the neighborhood of the contact point by functions

$$u_1(\phi_1), u_2(\phi_1), \psi_1(\phi_1), \dots, \phi_2(\phi_1) \quad (2.1.7)$$

These functions are of class  $C^1$  (at least they have continuous derivatives of the first order). Functions (2.1.7) and equations (2.1.5) enable calculation of the transmission errors (deviation of  $\phi_2(\phi_1)$  from the prescribed linear function) and the path of the contact point over the gear tooth surface.

For the case when the gear tooth surface is a regular screw involute surface, it may be directly represented in a two-parametric form and the number of equations in system (2.1.5) may be reduced to six.

## 2.2 Simulation of Contact

Due to the elastic approach of the gear tooth surfaces their contact is spread over an elliptical area. It is assumed that the magnitude of the elastic approach is known from experiments or may be predicted. Knowing in addition the principal curvatures and directions for two contacting surfaces at their point of contact we may determine the dimensions and orientation of the contact ellipse (Litvin, 1968).

The determination of principal curvature and directions for a surface represented in a three-parametric form is a complicated computational problem. A substantial simplification of this problem may be achieved using the relations between principal curvatures and directions, and the parameters of motion for two surfaces being in contact at a line. One of the contacting surfaces is the tool surface and the other is the generated surface.

Helical gears with modified gear tooth surfaces will be designed as surfaces being in point contact at every instant. The point of contact traces out on the surface a spatial curve (the path of contact) whose location must be controlled. The tangent to the path of contact and the derivative of the gear ratio  $\frac{d}{d\phi_1}(m_{21}(\phi_1))$  may be controlled by using the relationship between principal curvatures and directions for two surfaces that are in point contact (ref. 2). Here:

$$m_{21} = \frac{\omega_2}{\omega_1} = f(\phi_1)$$

is the gear ratio.

### 2.3 Partial Compensation of Transmission Errors

Aligned gears transform rotation with a constant gear ratio  $m_{21}$  and

$$\phi_{20}(\phi_1) = \frac{N_1}{N_2} \phi_1 \quad (2.3.1)$$

is a linear function. Here:  $N_1$  and  $N_2$  are the numbers of gear teeth. An investigation of the effect of helical gear rotational axis intersection or crossing indicates that  $\phi_2(\phi_1)$  becomes a piece-wise function which is nearly linear for each cycle of meshing (Fig. 2.2(a)). The transmission errors are determined by

$$\Delta\phi_2(\phi_1) = \phi_2(\phi_1) - \phi_1 \frac{N_1}{N_2} \quad (2.3.2)$$

and they are also represented by a piece-wise linear function (Fig. 2.2(b)). Transmission errors of this type cause a discontinuity of the gear angular velocity at transfer points and vibration becomes inevitable. The new topology of gear tooth surfaces proposed in this report allows the absorption of a linear function of transmission errors that results in a reduced level of vibration. This is based on the possibility to absorb a linear function by a parabolic function.

Consider the interaction of a parabolic function given by

$$\Delta\phi_2^{(1)} = -a\phi_1^2 \quad (2.3.3)$$

with a linear function represented by

$$\Delta\phi_2^{(2)} = b\phi_1 \quad (2.3.4)$$

The resulting function

$$\Delta\phi_2 = b\phi_1 - a\phi_1^2 \quad (2.3.5)$$

may be represented in a new coordinate system (Fig. 2.3):

$$\psi_2 = -a\psi_1^2 \quad (2.3.6)$$

where

$$\psi_2 = \Delta\phi_2 - \frac{b^2}{4a} \phi_1^2, \quad \psi_1 = \phi_1 - \frac{b}{2a} \phi_1^2 \quad (2.3.7)$$

We consider that  $\Delta\phi_2^{(1)} = -a\phi_1^2$  is a predesigned function that exists even if misalignments do not appear. The absorption of function  $\Delta\phi_2^{(2)} = b\phi_1$  by the parabolic function  $\Delta\phi_2^{(1)} = -a\phi_1^2$  means that gear misalignment does not change the predesigned parabolic function of transmission errors. Thus the resulting function of transmission errors  $\Delta\phi_2 = \Delta\phi_2^{(1)} + \Delta\phi_2^{(2)}$  will keep its shape as a parabolic function although the gears are misaligned. The resulting function of transmission errors  $\phi_2(\phi_1)$  may be obtained by translation of the parabolic function  $\Delta\phi_2^{(1)}$ .

The absorption of a linear function of transmission errors by a parabolic function is accompanied by the change of transfer

points. The transfer points determine the positions of the gears where one pair of teeth is rotating out of mesh and the next pair is coming into mesh. The change of transfer points is determined with  $\Delta\phi_1 = \left| \frac{b}{2a} \right|$  and  $\Delta\phi_2 = \frac{b^2}{4a^2}$ , the cycle of meshing of one pair of teeth is  $\phi_1 = \frac{2\pi}{N_1}$  i  $\in 1, 2$ . It may happen that the absorption of a linear function by a parabolic function is accompanied with a change that is too large. If this occurs the transfer points and the resulting parabolic function of transmission errors,  $\psi_2(\psi_1)$ , will be represented as a discontinuous function for one cycle of meshing (Fig. 2.4). To avoid this, it is necessary to limit the tolerances for gear misalignment.

#### 2.4 Misalignment of Regular Helical Gears

Regular helical pinion and gear can be represented by their surface position vectors and normal vectors in coordinate system  $S_1$  and  $S_2$  as:

$$[r_1] = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{\cos^2 \psi_n}{\cos \psi_c} \cos(\phi_p - \psi_c) [t_p \sin \beta_p + r_1 \phi_p] + r_1 \sin \phi_p \\ \frac{\cos^2 \psi_n}{\cos \psi_c} \sin(\phi_p - \psi_c) [t_p \sin \beta_p + r_1 \phi_p] + r_1 \cos \phi_p \\ t_p (\cos \beta_p + \sin^2 \psi_n \frac{\sin^2 \beta_p}{\cos \beta_p}) + r_1 \phi_p \sin^2 \psi_n t_g \beta_p \end{bmatrix} \quad (2.4.1)$$

$$[n_1] = \begin{bmatrix} n_{1x} \\ n_{1y} \\ n_{1z} \end{bmatrix} = \begin{bmatrix} \cos \psi_n \cos \beta_p \cos \phi_p + \sin \psi_n \sin \phi_p \\ -\cos \psi_n \cos \beta_p \sin \phi_p + \sin \psi_n \cos \phi_p \\ + \cos \psi_n \sin \beta_p \end{bmatrix}$$

(2.4.2)

$$[r_2] = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{\cos^2 \psi_n}{\cos \psi_c} \sin(\phi_G - \psi_c)[-t_G \sin \beta_p + r_2 \phi_G] + r_2 \sin \phi_G \\ \frac{\cos^2 \psi_n}{\cos \psi_c} \sin(\phi_G - \psi_c)[-t_G \sin \beta_p + r_2 \phi_G] + r_2 \cos \phi_G \\ t_G(\cos \beta_p + \sin^2 \psi_n \frac{\sin^2 \beta_p}{\cos \beta_p}) - r_2 \phi_G \sin^2 \psi_n \tan \beta_p \\ 1 \end{bmatrix} \quad (2.4.3)$$

$$[n_2] = \begin{bmatrix} n_{2x} \\ n_{2y} \\ n_{2z} \end{bmatrix} = \begin{bmatrix} \cos \psi_n \cos \beta_p \cos \phi_G + \sin \psi_n \sin \phi_G \\ -\cos \psi_n \cos \beta_p \sin \phi_G + \sin \psi_n \cos \phi_G \\ -\cos \psi_n \sin \beta_p \end{bmatrix} \quad (2.4.4)$$

where  $\psi_n$  and  $\psi_c$  are the pressure angles in gear tooth normal section and transverse section respectively;  $\beta_p$  is the helical angle at the pitch cylinder of the pinion and the gear;  $r_1$  and  $r_2$  are the radii pitch cylinder of the pinion and the gear respectively;  $\phi_p$  and  $t_p$  are the surface parameters of the pinion tooth surface;  $\phi_G$  and  $t_G$  are the surface parameters of the gear tooth surface.

When the helical pinion and gear are in mesh, with their axes misaligned, their position vectors and normal vectors can be transformed to the fixed coordinate system  $S_f$ . The basic ideas have already been discussed in the Section 2.1. And the real approach of transformation and the matrices to describe the misalignment and gear rotation will be given in Section 3.8.

It is found that when the misalignment occurs the regular

pinion and gear surface cannot contact in tangency. That is, in  $S_f$ , their normals can not be equal in all circumstances. In this case, only the gear tooth edge contacts pinion tooth surface in tangency. Therefore, in the fixed coordinate system, there are four equations to describe the contact, that is, the equalities of three position vectors describing gear tooth edge and pinion tooth surface as well as the zero product of gear tooth edge tangency and pinion tooth normal.

The computer aided simulation of meshing of misaligned helical gears with regular tooth surfaces shows that the transformation of rotation is accompanied with large transmission errors. There are two sub-cycles of meshing during the complete meshing cycle for one pair of teeth. These sub-cycles correspond to the meshing of (1) a curve with a surface, and (2) a point with the surface. The curve is the involute curve at the edge of the tooth of the gear and the point is the tip of the gear tooth edge. The transmission errors for the period of a cycle are represented by two linear functions (Fig. 2.5). The transformation of rotation will be accompanied with a jump of the angular velocity of the driven gear and therefore vibrations are inevitable.

The results of computation are presented for the following case. Given: number of teeth  $N_1 = 20$ ,  $N_2 = 40$ , diametral pitch in normal section  $P_n = 10 \text{ in}^{-1}$ , gear tooth length  $L = 10/P_n$  the helical angle  $\beta_p = 15^\circ$ , the normal pressure angle  $\psi_n = 20^\circ$ . The gear axes are crossed and form an angle  $\Delta\gamma = 5 \text{ arc minutes}$ . The computed transmission errors are represented in table 2.4.1.

TABLE 2.4.1 - TRANSMISSION ERRORS OF REGULAR HELICAL  
GEARS WITH CROSSED AXES

$\phi_1'$ deg	-8	-5	-2	1	4	7	10
$\Delta\phi_2'$ sec	4.90	3.06	1.22	-0.61	-2.45	-4.29	-6.12

## 2.5 Surface Deviation by the Change of Pinion Lead

Helical gears in this case are designed as helical gears with crossed axes. The crossing angle is chosen with respect to the expected tolerances of the gear misalignment ( $\Delta\gamma$  is in the range of 10 to 15 arc minutes). The gear ratio for helical gears with crossed axes may be represented (Litvin, 1968) as

$$M_{12} = \frac{\omega_1}{\omega_2} = \frac{r_{b2} \sin \lambda_{b2}}{r_{b1} \sin \lambda_{b1}} \quad (2.5.1)$$

where  $r_{bi}$  and  $\lambda_{bi}$  are the radius of the base cylinder and the lead angle on this cylinder,  $i \in 1, 2$ .  $|\lambda_{p2} - \lambda_{p1}| = \Delta\gamma$ . Here:  $\lambda_{pi}$  is the lead angle on the pitch cylinder. The advantage of application of crossed helical gears is that the gear ratio is not changed by the misalignment (by the change of  $\Delta\gamma$ ). The tooth surfaces contact each other at a point during meshing. The disadvantage of this type of surface deviation is that location of the bearing contact of the gears is very sensitive to gear misalignment. A slight change of the crossing angle causes shifting of the contact to the edge of the tooth (Fig. 2.6).

The discussed type of surface deviation is reasonable to apply for manufacturing of expensive reducers of large dimensions

when the lead of the pinion can be adjusted by regrinding. While changing by regrinding the parameters  $r_{b1}$  and  $\gamma_{b1}$ , the requirement that the product  $r_{b1} \sin \lambda_{b1}$  must be kept constant. Then, the gear ratio  $M_{21}$  will be of the prescribed value and transmission errors caused by the crossing of axes will be zero.

Theoretically, transmission errors are inevitable if the axes of crossed helical gears become intersected. Actually, if gear misalignment is of the range of 5 to 10 arc minutes, the transmission errors are very small and may be neglected. The main problem for this type of misalignment is again the shift of the bearing contact to the edge (Fig. 2.6).

### **3. GENERATION OF PINION TOOTH SURFACE BY A SURFACE OF REVOLUTION**

#### **3.1 Basic Consideration**

The purpose of this method for deviation of the pinion tooth surface is to reduce the sensitivity of the gears to misalignment. Also the transmission error must be kept to a low level and stabilize the bearing contact. This investigation shows that this goal may be achieved by the proposed method of crowning but the bearing contact cannot cover the whole surface. The reason for this is that the instantaneous contact ellipse moves across but not along the surface (Fig. 3.1).

The proposed method for generation is based on the following considerations. It is well known that the generation of a helical gear may be performed by an imaginary rack-cutter with skew teeth whose normal section represents a regular rack-cutter

for spur gears (Fig. 3.2(a)). We may imagine that two generating surfaces,  $\Sigma_G$  and  $\Sigma_p$ , are applied to generate the gear tooth surface and the pinion tooth surface, respectively (Fig. 3.2(b)). Surface  $\Sigma_G$  is a plane (a regular rack-cutter surface), and  $\Sigma_p$  is a cone surface. Surfaces  $\Sigma_G$  and  $\Sigma_p$  are rigidly connected and perform translational motion, while the pinion and the gear rotate about their axes (Fig. 3.3). The generated pinion and gear will be in point contact and transform rotation with the prescribed linear function  $\phi_2(\phi_1)$ . However, due to gear misalignment, function  $\phi_2(\phi_1)$  becomes a piecewise linear function (Fig. 2.2(a)) that is not acceptable. To absorb a linear function of transmission errors (Fig. 2.2(b)), a predesigned parabolic function of transmission errors is used. For this reason a surface of revolution that slightly deviates from the cone surface is proposed (Fig. 3.2(c)). The radius of the surface of revolution in its axial section determines the level of the predesigned parabolic function. The pinion crowning process may be accomplished by grinding, shaving or lapping.

### 3.2 Principle of Generation and Used Coordinate Systems

Consider two rigid connecting surfaces  $\Sigma_G$  and  $\Sigma_p$ . The generating surface  $\Sigma_G$  is a plane and generates the helical gear tooth surface that is an involute screw surface. Surface  $\Sigma_p$  is a surface of revolution. Initially, we consider that  $\Sigma_p$  is a cone surface and  $\Sigma_p$  and  $\Sigma_G$  contact each other along a straight line that is the generatrix of the cone.

Fig. 3.4 shows the generating surfaces  $\Sigma_G$  and  $\Sigma_p$ . Fig. 3.5 illustrates the process for generation. While the rigidly connected generating surfaces perform a translational motion, the pinion and gear rotate about their axes  $O_1$  and  $O_2$ , respectively. The parameter of motion of the cutter,  $S$ , and the angles of rotation of the pinion and the gear,  $\phi_1$  and  $\phi_2$ , are related as follows:

$$S = r_1 \phi_p = r_2 \phi_G \quad (3.2.1)$$

where  $r_1$  and  $r_2$  are the great centrodes radii, the cutter centode is the straight line that is tangent to the gear centrodes. Point I is the instantaneous center of rotation. Coordinate systems  $S_1$  and  $S_2$  are rigidly connected to the helical pinion and the helical gear, whereas coordinate systems  $S_c$  and  $S_f$  are rigidly connected to the tool surface and fixed frame. The generating surface  $\Sigma_G$  (a plane) and  $\Sigma_p$  are covered with a set of contact lines  $L_{G2}$  and  $L_{p1}$  respectively---the instantaneous lines of tangency of surfaces  $\Sigma_G$  and  $\Sigma_2$  and  $\Sigma_p$  and  $\Sigma_1$  (Fig. 2.3, a,b). The location of these lines depends on the value of parametric  $\phi_G$  and  $\phi_p$  ( $\phi_p$  and  $\phi_G$  are related). Line  $L_{Gp}$  is the line of tangency of generating surfaces  $\Sigma_G$  and  $\Sigma_p$ . When  $\Sigma_1$  and  $\Sigma_2$  are generated, at any moment, one point  $M_i$  on line  $L_{pG}$  are generating corresponding point  $M_{pi}$  and  $M_{Gi}$  on the helical gear surface  $\Sigma_1$  and  $\Sigma_2$ . When the  $\Sigma_1$  and  $\Sigma_2$  are meshing without misalignment  $M_{pi}$  and  $M_{Gi}$  contact each other in turn. Now it is clear why  $L_{pG}$  is not parallel with the edge of  $\Sigma_G$ . The reason is

that in this way, the contact ratio of crowned helical pinion and regular helical gear will be higher.

### 3.3 Tool Surface

The pinion generating surface is a cone and may be represented in an auxiliary coordinate system  $S_d$  (Fig. 3.7) as follows:

$$\tilde{x}_d = \begin{bmatrix} x_d \\ y_d \\ z_d \\ 1 \end{bmatrix} = \begin{bmatrix} u_p \cos\theta \sin\alpha \\ d - u_p \cos\alpha \\ -u_p \sin\theta_p \sin\alpha \\ 1 \end{bmatrix} \quad (3.3.1)$$

where  $0 < u < \frac{d}{\cos\alpha}$ ,  $0 < \theta < 2\pi$ . The surface normal is represented by

$$\tilde{N}_d = \frac{\partial \tilde{x}_d}{\partial \theta_p} \times \frac{\partial \tilde{x}_d}{\partial u_p} = u_p \sin\alpha \begin{bmatrix} \cos\alpha \cos\theta_p \\ \sin\alpha \\ \cos\alpha \sin\theta_p \end{bmatrix} \quad (3.3.2)$$

The unit surface normal is (provided  $u_p \sin\alpha \neq 0$ )

$$\tilde{n}_d = \begin{bmatrix} \cos\alpha \cos\theta_p \\ \sin\alpha \\ \cos\alpha \sin\theta_p \end{bmatrix} \quad (3.3.3)$$

Figure 3.8 illustrates the installment of the conic tool in coordinate system  $S_c$  step by step as follows:

(i) From coordinate system  $S_d$  to  $S_b'$ , the cone is tangent to the plane  $y_b'$ ,  $o_b'$ ,  $z_b'$  where the tangent line  $L_{pG}$  (see Section 3.2) is coincident with  $y_b'$  axis. Here, we have (Fig. 3.8a).

$$[M_{b'd}] = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & -d\sin\alpha \\ -\sin\alpha & \cos\alpha & 0 & d\operatorname{tg}\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3.4)$$

where the  $d$  is the height of cone.

(ii) From coordinate system  $S_b'$  to  $S_b$  the tangent line of cone and plane  $y_b o_b z_b$  is declined with an angle  $\mu$  (see Fig. 3.4), also the origin  $o_b'$  and  $o_b$  are not coincident. Here we have (Fig. 3.8b).

$$[M_{bb'}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\mu & -\sin\mu & -\frac{b}{\cos\psi_n} \\ 0 & \sin\mu & \cos\mu & \frac{b}{\cos\psi_n} \operatorname{tg}\mu \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3.5)$$

where  $b$  is the addendum height of the tool.

(iii) From coordinate system  $S_b$  to  $S_a$ , the tool is declined with a pressure angle  $\psi_n$ . For the helical gear, the  $\psi_n$  is measured at the normal cross section. Here, we have (Fig. 3.8c).

$$[M_{ab}] = \begin{bmatrix} \cos\psi_n & -\sin\psi_n & 0 & 0 \\ \sin\psi_n & \cos\psi_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3.6)$$

(iv) From coordinate system  $S_a$  to  $S_c$ , the helix angle  $\beta_p$  is considered. Here we have (Fig. 3.8d)

$$[M_{ca}] = \begin{bmatrix} \cos\beta_p & 0 & -\sin\beta_p & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta_p & 0 & \cos\beta_p & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3.7)$$

In coordinate system  $S_c$ , the tool surface and its normal can be represented as:

$$[r_c^{(p)}] = [M_{ca}] [M_{ab}] [M_{bb}] [M_{bd}] [r_d] \quad (3.3.8)$$

$$[n_c^{(p)}] = [L_{ca}] [L_{ab}] [L_{bb}] [L_{bd}] [n_d]$$

where  $3 \times 3$  L matrix is from corresponding  $4 \times 4$  M matrix, excluding the last row and last column. Substituting Eq. (3.3.1), (3.3.3), (3.3.4), (3.3.5), (3.3.6) and (3.3.7) into Eq. (3.3.8), finally we obtain the tool surface  $\Sigma_p$  and its normal in coordinate system  $S_c$  as

$$[r_c^{(p)}] = \begin{bmatrix} x_c^{(p)} \\ y_c^{(p)} \\ z_c^{(p)} \\ 1 \end{bmatrix}, \quad [n_c^{(p)}] = \begin{bmatrix} n_{cx}^{(p)} \\ n_{cy}^{(p)} \\ n_{cz}^{(p)} \end{bmatrix} \quad (3.3.9)$$

where

$$\begin{aligned} x_c^{(p)} = & u_p \cos \theta_p \sin \alpha (\cos \alpha \cos \psi_n \cos \beta_p + \sin \alpha \sin \psi_n \cos \mu \cos \beta_p \\ & + \sin \alpha \sin \mu \sin \beta_p) + u_p \sin \theta_p \sin \alpha (\sin \psi_n \sin \mu \cos \beta_p - \cos \mu \sin \beta_p) \\ & + u_p \cos \alpha (\cos \alpha \sin \psi_n \cos \mu \cos \beta_p - \sin \alpha \cos \psi_n \cos \beta_p + \cos \alpha \sin \mu \sin \beta_p) \\ & - \left( \frac{d}{\cos \alpha} - \frac{b}{\cos \psi_n \cos \mu} \right) [\sin \psi_n \cos \mu \cos \beta_p + \sin \mu \sin \beta_p] \end{aligned}$$

$$\begin{aligned} y_c^{(p)} = & u_p \cos \theta_p \sin \alpha (\cos \alpha \sin \psi_n - \sin \alpha \cos \psi_n \cos \mu) \\ & - u_p \sin \theta_p \sin \alpha \cos \psi_n \sin \mu - u \cos \alpha (\sin \alpha \sin \psi_n + \cos \alpha \cos \psi_n \cos \mu) \\ & + \left( \frac{d}{\cos \alpha} - \frac{b}{\cos \psi_n \cos \mu} \right) \cos \mu \cos \psi_n \end{aligned}$$

$$\begin{aligned}
z_c^{(p)} = & u_p \cos \theta_p \sin \alpha (\cos \alpha \cos \psi_n \sin \beta_p + \sin \alpha \sin \psi_n \cos \mu \sin \beta_p - \sin \alpha \sin \mu \cos \beta_p) \\
& + u_p \sin \theta_p \sin \alpha (\sin \psi_n \sin \mu \sin \beta_p + \cos \mu \cos \beta_p) \\
& + u_p \cos \alpha (\cos \alpha \sin \psi_n \cos \mu \sin \beta_p - \sin \alpha \cos \psi_n \sin \beta_p - \cos \alpha \sin \mu) \\
& + \left( \frac{p_d}{\cos \alpha} - \frac{b}{\cos \psi_n \cos \mu} \right) (-\cos \mu \sin \beta_p \sin \psi_n + \sin \mu \cos \beta_p)
\end{aligned}$$

$$\begin{aligned}
n_{cx}^{(p)} = & \cos \theta_p \cos \alpha (\cos \alpha \cos \psi_n \cos \beta_p + \sin \alpha \sin \psi_n \cos \mu \cos \beta_p \\
& + \sin \alpha \sin \mu \sin \beta_p) + \sin \theta_p \cos \alpha (\sin \psi_n \sin \mu \cos \beta_p - \cos \mu \sin \beta_p) \\
& + \sin \alpha (\sin \alpha \cos \psi_n \cos \beta_p - \cos \alpha \sin \psi_n \cos \mu \cos \beta_p - \cos \alpha \sin \mu \sin \beta_p)
\end{aligned}$$

$$\begin{aligned}
n_{cy}^{(p)} = & \cos \theta_p \cos \alpha (\cos \alpha \sin \psi_n - \sin \alpha \cos \psi_n \cos \mu) - \sin \theta_p \cos \alpha \cos \psi_n \sin \beta_p \\
& + \sin \alpha (\sin \alpha \sin \psi_n + \cos \alpha \cos \psi_n \cos \mu)
\end{aligned}$$

$$\begin{aligned}
n_{cz}^{(p)} = & \cos \theta_p \cos \alpha (\cos \alpha \cos \psi_n \sin \beta_p + \sin \alpha \sin \psi_n \cos \mu \sin \beta_p - \sin \alpha \sin \mu \cos \beta_p) \\
& + \sin \theta_p \cos \alpha (\sin \psi_n \sin \mu \sin \beta_p + \cos \mu \cos \beta_p) \\
& + \sin \alpha (\sin \alpha \cos \psi_n \sin \beta_p - \cos \alpha \sin \psi_n \cos \mu \sin \beta_p + \cos \alpha \sin \mu \cos \beta_p)
\end{aligned}$$

### 3.4 Pinion Tooth Surface

The equation of meshing of the generating surface  $\Sigma_p$  and the helical pinion tooth surface is represented by

$$\tilde{N}^{(P)} \cdot \tilde{V}^{(Pl)} = \tilde{N}^{(P)} \cdot (\tilde{V}^{(P)} - \tilde{V}^{(l)}) = 0 \quad (3.4.1)$$

We can also use equation based on the fact that the contact normal of generating and generated surfaces must intersect the instantaneous axis of rotation I-I (Litvin, 1968). Thus, we obtain (Fig. 2.2).

$$\frac{x_c - x_c^{(P)}}{n_{cx}^{(P)}} = \frac{y_c - y_c^{(P)}}{n_{cy}^{(P)}} = \frac{z_c - z_c^{(P)}}{n_{cz}^{(P)}} \quad (3.4.2)$$

where  $(X_c, Y_c, Z_c)$  are the coordinates of a point that lies on axis I-I;  $x_c^{(P)}, y_c^{(P)}, z_c^{(P)}$  are the coordinates of cone surface;  $n_{cx}, n_{cy}$  and  $n_{cz}$  are the projections of the surface unit normal. From Fig. 3.5, it is known that

$$X_c = S = r_1 \phi_p, \quad Y_c = 0$$

Equation (3.4.2), (3.4.3), and (3.3.9) yield

$$\begin{aligned} f_1(u_p, \theta_p, \phi_p) = & -u_p [\cos \theta_p (\cos \mu \cos \beta_p + \sin \mu \sin \psi_n \sin \beta_p) \\ & + \sin \theta_p (\sin \alpha \sin \mu \cos \beta_p - \cos \alpha \cos \psi_n \sin \beta_p - \sin \alpha \sin \psi_n \cos \mu \sin \beta_p)] \\ & + \left( \frac{d}{\cos \alpha} - \frac{b}{\cos \psi_n \cos \mu} \right) [(\cos \theta_p \cos^2 \alpha + \sin^2 \alpha) \cdot [\cos \mu \cos \beta_p \\ & + \sin \psi_n \sin \mu \sin \beta_p] - \sin \theta_p \cos \alpha \cos \psi_n \sin \beta_p] \\ & + r \phi_p [\cos \theta_p \cos \alpha (\cos \alpha \sin \psi_n - \sin \alpha \cos \psi_n \cos \mu) \\ & + \sin \alpha (\sin \alpha \sin \psi_n + \cos \alpha \cos \psi_n \cos \mu) - \sin \theta_p \cos \alpha \cos \psi_n \sin \psi_n] \\ & = 0 \end{aligned} \quad (3.4.4)$$

The helical pinion tooth surface can be obtained by transforming the generating tool surface  $\Sigma_p$  to coordinate system  $S_1$ , together with equation of meshing. The coordinate transformation in transition from  $S_c$  to  $S_f$  is represented by the matrix  $M_{fc}$  as (Fig. 3.5)

$$[M_{fc}] = \begin{bmatrix} 1 & 0 & 0 & -r_1 \phi_p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4.5)$$

The coordinate transformation in transition from  $S_f$  to  $S_1$  is represented by the matrix  $M_{1f}$  as (Fig. 3.5).

$$[M_{1f}] = \begin{bmatrix} \cos\phi_p & \sin\phi_p & 0 & 0 \\ -\sin\phi_p & \cos\phi_p & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4.6)$$

Therefore, the helical pinion surface can be represented in coordinate system  $S_1$  as:

$$[r_1] = [M_{1c}][r_c^{(P)}] = [M_{1f}][M_{fc}][r_c^{(P)}] \quad (3.4.7)$$

together with Eq. (3.4.4). Substituting Eq. (3.4.5), (3.4.6) and (3.3.9) into Eq. (3.4.7) we have

$$[r_1] = \begin{bmatrix} (x_c^{(P)} - r_1\phi_p)\cos\phi_p + y_c^{(P)}\sin\phi_p \\ (x_c^{(P)} - r_1\phi_p)\sin\phi_p + y_c^{(P)}\cos\phi_p \\ z_c^{(P)} \\ 1 \end{bmatrix} \quad (3.4.8)$$

Eq. (3.4.8) and (3.4.4) represents the pinion tooth surface where  $x_c^{(P)}$ ,  $y_c^{(P)}$  and  $z_c^{(P)}$  are expressed in Eq. (3.3.9).

Using the same approach, the unit normal pinion tooth surface can be represented by Eq. (3.4.9) and (3.4.4) where  $n_{cx}^{(P)}$ ,  $n_{cy}^{(P)}$  and  $n_{cz}^{(P)}$  are expressed in Eq. (3.3.9)

$$[n_1] = \begin{bmatrix} n_{cx}^{(P)}\cos\phi_p + n_{cy}^{(P)}\sin\phi_p \\ -n_{cx}^{(P)}\sin\phi_p + n_{cy}^{(P)}\cos\phi_p \\ n_{cz}^{(P)} \end{bmatrix} \quad (3.4.9)$$

It is clear that if we set  $\beta_p = 0$ , the pinion becomes spur pinion. Therefore, crowning spur pinion using conic tool is only a special case of crowning helical pinion.

### 3.5 Condition of Pinion Non-Undercutting

The problem of undercutting of the helical pinion tooth surface by crowning is related with the appearance on the pinion tooth surface of singular points. From differential geometry, it is known that the surface point is singular if the surface normal is equal to zero at such a point.

Litvin, proposed a method to determine a line on the tool surface whose points will generate singular points on the surface generated by the tool. This line designated by L (Fig. 3.9) must be out of the working part of the tool surface to avoid undercutting of the pinion by crowning.

The limiting line L of the tool surface is determined by the following equations

$$\tilde{r}_c^{(P)} = \tilde{r}_c^{(P)}(u_p, \theta_p) \quad (3.5.1)$$

$$f_1(u_p, \theta_p, \phi_p) = 0 \quad (3.5.2)$$

$$F(u_p, \theta_p, \phi_p) = 0 \quad (3.5.3)$$

Vector equation (3.5.1) represents the tool surface [see Eq. (3.3.9)]; equation (3.5.2) is the equation of meshing [see Eq. (3.4.4)] and equation (3.5.3) comes from the requirement of

limiting line L as:

$$\begin{vmatrix} \frac{\partial X_c^{(P)}}{\partial u_p} & \frac{\partial X_c^{(P)}}{\partial \theta_p} & v_{cx}^{(cl)} \\ \frac{\partial Z_c^{(P)}}{\partial u_p} & \frac{\partial Z_c^{(P)}}{\partial \theta_p} & v_{cz}^{(cl)} \\ \frac{\partial f_1}{\partial u_p} & \frac{\partial f_1}{\partial \theta_p} & \frac{\partial f_1}{\partial \phi} \frac{\partial \phi}{\partial f} \end{vmatrix} = 0 \quad (3.5.4)$$

where  $\tilde{v}_c^{(cl)}$  is the relative velocity of the tool and generated surface at generating point, represented in coordinate system  $S_c$ . Actually, we can write  $\tilde{v}_c^{(cl)}$  as

$$\tilde{v}_c^{(cl)} = \tilde{v}_c^{(c)} - \tilde{v}_c^{(1)} \quad (3.5.5)$$

where  $\tilde{v}_c^{(c)}$  is the velocity of the cutter and  $\tilde{v}_c^{(1)}$  is the velocity of the pinion. From Fig. 2.2, we get

$$\tilde{v}_c^{(c)} = \begin{bmatrix} -\frac{ds}{dt} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \omega_p \\ 0 \\ 0 \end{bmatrix} \quad (3.5.6)$$

$$\tilde{v}_c^{(1)} = \tilde{\omega}_p \times \tilde{r}_c^{(p)} + \overline{O_c O_1} \times \tilde{\omega}_p \quad (3.5.7)$$

where

$$\tilde{\omega}_p = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \omega \quad \overline{O_c O_1} = \begin{bmatrix} r_1 \phi_p \\ -r_1 \\ 0 \end{bmatrix} \quad (3.5.8)$$

Equations (3.5.6), (3.5.7) and (3.5.8) yield

$$\tilde{y}_c^{(cl)} = \begin{bmatrix} y_c^{(p)} \\ -x_c^{(p)} + r_1 \phi_p \\ 0 \end{bmatrix} \omega \quad (3.5.9)$$

where  $x_c^{(p)}$  and  $y_c^{(p)}$  are represented by equations 3.3.9.

Equation (3.5.9), (3.3.9), (3.4.3) and (3.5.4) yield

$$u_p^2 YW + u_p(r_1 G^3 + u_p^*(WZ + XY)) + u_p^{*2} XZ = 0 \quad (3.5.10)$$

where

$$\begin{aligned} W = & (B^2 E + A^2 E) + (BD - AF) I + \sin^3 \theta_p (A^2 F - B^2 F + 2ABD) \\ & - \cos^3 \theta_p (2AFB + B^2 D - A^2 D) - \sin \theta_p (2A^2 F + ABD - AEI) \\ & + \cos \theta_p [BEI + 2B^2 D + ABF] \end{aligned}$$

$$\begin{aligned} X = & [(AF \sin^2 \alpha - CE) \cos \theta_p + (AD \sin^2 \alpha - AE \cos^2 \alpha) \sin \theta_p \\ & + (AF \cos^2 \alpha - DC)] (B \cos \theta_p + A \sin \theta_p + I) \end{aligned}$$

$$Y = D \tan \alpha \cos \theta_p - F \tan \alpha \sin \theta_p - E \tan \alpha$$

$$Z = \cos \mu \cos \psi_n$$

$$U_p^* = \left( \frac{d}{\cos \alpha} - \frac{b}{\cos \psi_n \cos \mu} \right)$$

$$A = \cos \mu \cos \beta_p + \sin \psi_n \sin \mu \sin \beta_p$$

$$B = \cos \alpha \cos \psi_n \sin \beta_p + \sin \alpha \sin \psi_n \cos \mu \sin \beta_p - \sin \alpha \sin \mu \cos \beta_p$$

$$C = \cos \alpha \cos \psi_n \sin \beta_p$$

$$D = (\cos \alpha \sin \psi_n - \sin \alpha \cos \psi_n \cos \mu) \cos \alpha$$

$$E = (\sin \alpha \sin \psi_n + \cos \alpha \cos \psi_n \cos \mu) \sin \alpha$$

$$F = \cos \alpha \cos \psi_n \sin \mu$$

$$G = D \cos \theta + E - F \sin \theta$$

$$I = \operatorname{ctg} \alpha (\cos \alpha \sin \psi_n \cos \mu \sin \beta_p - \sin \alpha \cos \psi_n \sin \beta_p - \cos \alpha \sin \mu \cos \beta_p)$$

Using Eq. (3.5.10) and taking it into account that  $\frac{\partial F}{\partial \theta_p} \neq 0$ , we can represent the  $u_p$  as a function of  $\theta_p$ . Undercutting will be avoided if

$$u_p(\theta_p) > \frac{d}{\cos \alpha} \quad (3.5.11)$$

The analysis of Eq. (3.5.10) indicates that the inequality (3.5.11) is satisfied if the condition of helical pinion non-

undercutting by a regular rack cutter is satisfied.

### 3.6 Principal Directions and Curvatures of Tooth Surface

A simplified approach to determine principal directions and curvatures of helical pinion has been proposed by Litvin (Litvin, 1968). The main idea is representing the principal directions and curvatures of generated surface  $\Sigma_1$  by the principal directions and curvatures of generating surface  $\Sigma_p$ .

Let us determine the principal directions of curvatures of tool surface  $\Sigma_p$ . The tool surface is a cone surface and its principal directions coincide with the direction of the cone generatrix and the direction that is perpendicular to the cone generatrix.

The Rodrigue's formula (Eq. 3.6.1) can be used to obtain the principal curvature and directions

$$\kappa_{I,II} \tilde{V}_r = -\dot{\tilde{n}}_r \quad (3.6.1)$$

where  $\tilde{V}_r$  is the velocity of a point that moves over a surface and  $\dot{\tilde{n}}_r$  is the derivative of the surface unit normal  $\tilde{n}$ , when  $\tilde{n}$  changes its direction due to the motion over the surface.

Using Eq. 3.6.1, the principal directions and curvatures of cone surface  $\Sigma_p$  can be expressed in coordinate system  $S_d$  as:

$$\mathbf{e}_I^{(P)} = \frac{\partial \mathbf{r}_d}{\partial \theta_p} \div \left| \frac{\partial \mathbf{r}_d}{\partial \theta_p} \right| = \begin{bmatrix} -\sin\theta \\ 0 \\ \cos\theta \end{bmatrix}, \quad \kappa_I^{(P)} = -\frac{1}{u_p \cos\alpha}$$

$$\mathbf{e}_{II}^{(P)} = \frac{\partial \mathbf{r}_d}{\partial u_p} \div \left| \frac{\partial \mathbf{r}_d}{\partial u_p} \right| = \begin{bmatrix} \cos\theta \sin\alpha \\ -\cos\alpha \\ \sin\theta \sin\alpha \end{bmatrix}, \quad \kappa_{II}^{(P)} = 0 \quad (3.6.2)$$

The negative sign for  $\kappa_I$  indicates that the curvature center is located on the negative direction of the surface normal. The principal curvatures are invariants with respect to the used coordinate system. Whereas the principal direction will be represented in coordinate system  $S_f$  as

$$[\mathbf{e}_{I,II}^{(P)}]_f = [L_{fc}] [L_{cd}] [\mathbf{e}_{I,II}^{(P)}]_d \quad (3.6.3)$$

where  $[L_{cd}] = [L_{ca}] [L_{ab}] [L_{ab}'] [L_{b'd}]$ , 3 x 3 L matrices can be obtained from corresponding 4 x 4 M matrices [see Eq. (3.3.4), (3.3.5), (3.3.6) and (3.3.7)],  $L_{fc}$  is 3 x 3 unitary matrix (see Fig. 3.5).

The determination of principal curvatures and directions for the pinion tooth is based on the following equations (see Litvin, 1968)

$$\tan 2\sigma^{(p1)} = \frac{2b_{13}b_{23}}{b_{23}^2 - b_{13}^2 - (\kappa_I^{(p)} - \kappa_{II}^{(p)})b_{33}} \quad (3.6.4)$$

$$\kappa_{II}^{(1)} - \kappa_I^{(1)} = \frac{b_{23}^2 - b_{13}^2 - (\kappa_I^{(p)} - \kappa_{II}^{(p)})b_{33}}{b_{33} \cos 2\sigma^{(p1)}} \quad (3.6.5)$$

$$\kappa_{II}^{(1)} + \kappa_I^{(1)} = \kappa_I^{(p)} + \kappa_{II}^{(p)} + \frac{b_{13}^2 + b_{23}^2}{b_{33}} \quad (3.6.6)$$

where  $\kappa_I^{(p)}$ ,  $\kappa_{II}^{(p)}$  and  $\underline{e}_I^{(p)}$ ,  $\underline{e}_{II}^{(p)}$  are the principal curvatures and unit vector of principal directions of  $\Sigma_p$ .  $\kappa_I^{(1)}$ ,  $\kappa_{II}^{(2)}$  and  $\underline{e}_I^{(1)}$ ,  $\underline{e}_{II}^{(1)}$  are the principal curvatures and unit vectors of principal directions of  $\Sigma_p$ . Angle  $\sigma^{(p1)}$  is measured counter clockwise from  $\underline{e}_I^{(p)}$  to  $\underline{e}_I^{(1)}$  (Fig. 3.10). The coefficient  $b_{13}$ ,  $b_{23}$  and  $b_{33}$  have been derived as shown in (Litvin 1968) but modified for the case when a rack cutter generates a gear. The expressions for  $b_{13}$ ,  $b_{23}$ , and  $b_{33}$  are as follows:

$$\begin{bmatrix} b_{13} \\ b_{23} \end{bmatrix} = \begin{bmatrix} \underline{e}_{II}^{(p)} \cdot \underline{\omega}_p \\ -\underline{e}_I^{(p)} \cdot \underline{\omega}_p \end{bmatrix} - \begin{bmatrix} -\kappa_I^{(p)} & 0 \\ 0 & -\kappa_{II}^{(p)} \end{bmatrix} \begin{bmatrix} \underline{v}^{(p1)} \cdot \underline{e}_I^{(p)} \\ \underline{v}^{(p1)} \cdot \underline{e}_{II}^{(p)} \end{bmatrix} \quad (3.6.7)$$

$$\begin{aligned} b_{33} = & [\underline{n} \cdot \underline{\omega}_p \underline{v}_{tr}^{(p)}] + [\underline{n} \cdot \underline{\omega}_p \underline{v}^{(p1)}] - \kappa_I^{(p)} (\underline{v}^{(p1)} \cdot \underline{e}_I^{(p)})^2 \\ & - \kappa_{II}^{(p)} (\underline{v}^{(p1)} \cdot \underline{e}_{II}^{(p)})^2 - \frac{\omega^2}{m_{lp}^2} m'_{lp} [\underline{n} \cdot \underline{x}^{(p1)} \underline{k}_f] \end{aligned} \quad (3.6.8)$$

where  $m_{lp} = \frac{d\phi_p}{ds}$ ,  $m'_{lp} = \frac{dm_{lp}}{ds}$ .

The vectors used in Eq. (3.6.7) and (3.6.8) are represent in coordinate system  $S_f$  (Fig. 3.5) with  $\underline{i}_f$ ,  $\underline{j}_f$ ,  $\underline{k}_f$  as its unit vectors of axes. The expressions of the vectors in Eq. (3.6.7)

and (3.6.8) are as follows (Fig. 3.5)

$$\omega_p = \omega \frac{k_f}{z_f} \quad (3.6.9)$$

is the angular velocity of the pinion being in meshing with the rack cutter.

$$v_{tr}^{(p)} = -\omega r_1 \frac{1}{z_f} \quad (3.6.10)$$

is the transfer velocity of a point on the rack cutter that performs translational motion,  $r_1$  is the radius of pinion centreode.

$$v_{tr}^{(1)} = \omega_p \times r_f^{(p)} = \omega \begin{bmatrix} -y_f \\ x_f \\ 0 \end{bmatrix} = \omega \begin{bmatrix} -(y_c + r_1) \\ x_c - r_1 \phi_p \\ 0 \end{bmatrix} \quad (3.6.11)$$

is the transfer velocity of the pinion

$$v^{(p1)} = v^{(p)} - v^{(1)} = \omega \begin{bmatrix} y_c \\ -x_c + r_1 \phi_p \\ 0 \end{bmatrix} \quad (3.6.12)$$

is the "sliding" velocity - the velocity of a point of rack cutter with respect to the same point of pinion.

Substituting Eq. (3.6.9), (3.6.10), (3.6.11), (3.6.12), (3.6.2) and (3.3.9) into Eq. (3.6.7) and (3.6.8), then substituting Eq. (3.6.7) and (3.6.8) into (3.6.4), (3.6.5) and

(3.6.6), finally we can obtain the principal directions and curvatures from Eq. (3.6.4), (3.6.5), (3.6.6) and (3.6.2). Since the expressions of tool surface and its principal direction are complicated and tedious. It is difficult for us to write down the expression of  $\kappa_{II}^{(1)}$ ,  $\kappa_{II}^{(1)}$ ,  $\underline{e}_I^{(1)}$  and  $\underline{e}_{II}^{(2)}$ . However, a corresponding computer program is developed for determining the principal directions and curvatures of any point of the pinion surface using the algorithm discussed above.

The same approach can be used to determine the principal directions and curvatures of regular helical gear. However, in this case, the generating surface is a plane, and the problem becomes simpler.

### 3.7 Contact Ellipse and Bearing Contact

The dimensions and orientation of the instantaneous contact ellipse can be determined based on the equations proposed by Litvin. (Litvin 1968). The input data for computation is:

$\kappa_{I,II}^{(i)}$  ( $i = 1, 2$ ),  $\underline{e}_I^{(1)}$ ,  $\sigma^{(12)}$  and  $\epsilon$ , where  $\kappa_{I,II}^{(i)}$  are the principal curvature of pinion tooth surface  $\Sigma_1$  and gear tooth surface  $\Sigma_2$ .  $\underline{e}_I^{(1)}$  is the unit vector of the first principal direction of  $\Sigma_1$ .  $\sigma^{(12)}$  is the angle formed by the unit vectors of principal directions  $\underline{e}_I^{(1)}$  and  $\underline{e}_{II}^{(2)}$  (Fig. 3.11) and  $\epsilon$  is the elastic approach of the contacting surfaces. The bearing contact is formed by a set of instantaneous contact ellipses that move over the gear tooth surface in the process of meshing.

The axes of the contact ellipse are directed along the  $\eta$  axis and  $\xi$  axis, respectively. The orientation of contact

ellipse is determined by the angle  $\alpha$ , which is angle between  $\eta$  and  $e_I^{(1)}$ . (Fig. 3.11) The dimensions of ellipse are determined by  $a^*$  and  $b^*$  which are the half lengths of major and minor axis of the ellipse respectively. The following equations are used to determine  $\alpha$ ,  $a^*$  and  $b^*$ .

$$\begin{aligned} A &= \frac{1}{4} [\kappa_\epsilon^{(1)} - \kappa_\epsilon^{(2)} - |g_1 - g_2|] \\ B &= \frac{1}{4} [\kappa_\epsilon^{(1)} - \kappa_\epsilon^{(2)} + |g_1 + g_2|] \\ w \tan(2\alpha) &= \frac{g_1 \sin 2\sigma^{(12)}}{g_1 - g_2 \cos 2\sigma^{(12)}} \end{aligned} \quad (3.7.1)$$

where

$$\begin{aligned} \kappa_\epsilon^{(1)} &= \kappa_I^{(1)} + \kappa_{II}^{(2)}, \quad \kappa_\epsilon^{(2)} = \kappa_I^{(2)} + \kappa_{II}^{(2)} \\ g_1 &= \kappa_I^{(1)} - \kappa_{II}^{(1)}, \quad g_2 = \kappa_I^{(2)} - \kappa_{II}^{(2)} \end{aligned}$$

and

$$a^* = \left| \frac{\epsilon}{A} \right|^{1/2} \quad b^* = \left| \frac{\epsilon}{B} \right|^{1/2} \quad (3.7.2)$$

### 3.8 Simulation of Meshing and Determination of Transmission Errors

The simulation of meshing is a part of the computer aided tooth contact analysis (TCA) program. The simulation of meshing is based on equations that provide the continuous tangency of contacting surfaces.

To simulate the meshing of a crowned helical pinion and

regular involute helical gear with misaligned axis, we will use the following coordinate system, as shown in Fig. 3.11 and Fig. 3.12:

(i)  $S_f$  is rigidly connected to the frame (ii) an auxiliary coordinate system  $S_h$  that is also rigidly connected to the frame (iii)  $S_1$  and  $S_2$  are rigidly connected to the pinion and the gear respectively. The relations between  $S_1$  and  $S_f$  and between  $S_2$  and  $S_h$  are shown in Fig. 3.12 and expressed by  $M_{f1}$  and  $M_{h2}$  as

$$M_{f1} = \begin{bmatrix} \cos\phi_1 & \sin\phi_1 & 0 & 0 \\ -\sin\phi_1 & \cos\phi_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8.1)$$

$$M_{h2} = \begin{bmatrix} \cos\phi_2 & -\sin\phi_2 & 0 & 0 \\ \sin\phi_2 & \cos\phi_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8.2)$$

It is obvious that the rotation axis of the pinion and gear are  $Z_f(Z_1)$  and  $Z_h(Z_2)$  respectively. Therefore, the error of assembly of gears can be simulated with orientation and location of coordinate system  $S_h$  with respect to  $S_f$ . Figure 3.13 shows the orientation of  $S_h$  and  $S_f$ . When the pinion and gear axes are crossed (Fig. 3.13a) and intersected (Fig 3.13b) with operating center distance  $C = O_f O_h$ . It must be emphasized that  $C$  can be different from the nominal center distance given by  $C^0 = r_1 + r_2$ , where  $r_1$  and  $r_2$  are the radii of the pinion and gear pitch circles. The coordinate transformation from  $S_h$  to  $S_f$  is represented by  $[M_{fh}]$  as follows (Fig. 3.13)

$$[M_{fh}] = \begin{bmatrix} -\cos\Delta\gamma & 0 & \sin\Delta\gamma & 0 \\ 0 & -1 & 0 & C \\ \sin\Delta\gamma & 0 & \cos\Delta\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8.3)$$

where the gear axes are crossed as shown in Fig. 3.13a.

$$[M_{fh}] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -\cos\Delta_f & -\sin\Delta_f & C \\ 0 & -\sin\Delta_f & \cos\Delta_f & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8.4)$$

where the gear axes are intersected as shown in Fig. 3.13b.

The helical pinion tooth surface and its normal can be expressed in coordinate system  $S_f$  as:

$$\begin{aligned} [r_f^{(1)}] &= [M_{f1}][r_1] \\ [n_f^{(1)}] &= [L_{f1}][n_1] \end{aligned} \quad (3.8.5)$$

where  $L_{f1}$  is 3 x 3 matrix from  $M_{f1}$ . The helical gear tooth surface and its normal can be expressed in coordinate system  $S_f$  as

$$\begin{aligned} [r_f^{(2)}] &= [M_{fh}][M_{h2}][r_2] \\ [n_f^{(2)}] &= [L_{fh}][L_{h2}][r_2] \end{aligned} \quad (3.8.6)$$

where  $L_{fh}$  and  $L_{h2}$  is 3 x 3 matrices from  $M_{fh}$  and  $M_{h2}$ .

For a helical gear (lefthand),  $[r_2]$  and  $[n_2]$  can be represented as:

$$[r_2] = \begin{bmatrix} -\frac{\cos^2 \psi_n}{\cos \psi_c} \cos(\phi_G - \psi_n) [-t_G \sin \beta_p + r_2 \phi_G] + r_2 \sin \phi_G \\ \frac{\cos^2 \psi_c}{\cos \psi_c} \sin(\phi_G - \psi_c) [-t_G \sin \beta_p + r_2 \phi_G] + r_2 \cos \phi_G \\ t_G (\cos \beta_p + \sin^2 \psi_n \frac{\sin^2 \beta_p}{\cos \beta_p}) - r_2 \phi_G \sin^2 \psi_n \tan \beta_p \end{bmatrix} \quad (3.8.7)$$

$$[n_2] = \begin{bmatrix} -(\cos \psi_n \cos \beta_p \cos \phi_G + \sin \psi_n \sin \phi_G) \\ \cos \psi_n \cos \beta_p \sin \phi_G - \sin \psi_n \cos \phi_G \\ \cos \psi_n \sin \beta_p \end{bmatrix}$$

where  $t_G$  and  $\phi_G$  are the surface parameters and  $\psi_c = \arctan(tg \psi_n / \cos \beta_p)$  is the pressure angle in the transverse section of the gear tooth.

From Eq. (3.3.9), (3.4.4), (3.4.8), (3.8.5), (3.8.6) and (3.8.7), it is known that. [substituting Eq.(3.4.4) into(3.3.9) to eliminate  $u_p$ ]:

$$\begin{aligned} \tilde{r}_f^{(1)} &= \tilde{r}_f^{(1)}(\phi_p, \theta_p, \phi_1) \\ \tilde{n}_f^{(1)} &= \tilde{n}_f^{(1)}(\phi_p, \theta_p, \phi_1) \\ \tilde{r}_f^{(2)} &= \tilde{r}_f^{(2)}(\phi_G, t_G, \phi_2) \\ \tilde{n}_f^{(2)} &= \tilde{n}_f^{(2)}(\phi_G, t_G, \phi_2) \end{aligned} \quad (3.8.8)$$

The contact of gear tooth surface is simulated in the TCA program by the following equations

$$\tilde{r}_f^{(1)}(\phi_p, \theta_p, \phi_1) = \tilde{r}_f^{(2)}(\phi_G, t_G, \phi_2) \quad (3.8.9)$$

$$\tilde{n}_f^{(1)}(\phi_p, \theta_p, \phi_1) = \tilde{n}_f^{(2)}(\phi_G, t_G, \phi_2) \quad (3.8.10)$$

Vector equation (3.8.9) comes from the equality of the position vectors at the contact point and provide three independent equations whereas vector equation (3.8.10) comes from the equality of surface unit normals and provide only two independent equations. Therefore, Eq.(3.8.9) and (3.8.10) yield five independent scalar equations as follows:

$$f_i(\phi_1, \theta_p, \phi_p, \phi_2, t_G, \phi_G) = 0 \quad (i = 1, 2, \dots, 5) \quad (3.8.11)$$

Equation system 3.8.11 is the expression in implicit form of functions of one variable, i.e.  $\phi_1$ . Using theorem of implicit function, we can obtain

$$\phi_2 = \phi_2(\phi_1) \quad (3.8.12)$$

considering at the neighborhood of  $P^0 = (\phi_1, \theta_p, \phi_p, \phi_2, t_G, \phi_G)$  where  $P^0$  satisfy Eq.(3.8.11) and

$$J = \frac{D(f_1, f_2, f_3, f_4, f_5)}{D(\phi_p, \theta_p, \phi_2, t_G, \phi_G)} \neq 0 \quad (3.8.13)$$

The function of transmission error can be determined by

$$\Delta\phi_2(\phi_1) = \phi_2(\phi_1) - \frac{N_1}{N_2} \phi_1 \quad (3.8.14)$$

### 3.9 Modification of Generating Surface $\Sigma_p$

Using cone as a tool surface to generate the helical pinion is good way to localize bearing contact. By using the TCA program, it can be shown that the transmission errors caused by gear misalignment are on a very low level. However, the shape of the function of transmission errors is unfavorable as shown in Fig. 3.14(a). This shape of the function of transmission errors will result in interruption and interference with change of meshing gear teeth. A more favorable shape of the function of transmission errors is shown in Fig. 3.14(b). To obtain this shape of transmission error function, we need to modify generating surface  $\Sigma_p$  into a revolute surface. The reason for using revolute surface is similar to the case for crowning a spur pinion (see Litvin and Zhang, 1987).

The surface of revolution is generated by an arc circle with radius  $\rho$ . The arc has a common normal with the cone generatrix line at the point M (Fig. 3.15(a)). The circular arc and its normal are expressed in an auxiliary coordinate system  $S_e$  as follows

$$\begin{aligned} X_e &= \rho [\cos(\alpha + \beta) - \cos\alpha] + \left( \frac{d}{\cos\alpha} - \frac{b}{\cos\psi_n \cos\mu} \right) \sin\alpha \\ &= \left( \frac{d}{\cos\alpha} - \frac{b}{\cos\psi_n \cos\mu} \right) \sin\alpha - 2\rho \sin \frac{\beta}{2} \sin\left(\alpha + \frac{\beta}{2}\right) \\ Y_e &= \rho [\sin(\alpha + \beta) - \sin\alpha] + \frac{b}{\cos\psi_n \cos\mu} \cos\alpha \end{aligned} \quad (3.9.1)$$

$$= \frac{b}{\cos\psi_n \cos\mu} \cos\alpha + 2\rho \sin \frac{\beta}{2} \cos(\alpha + \frac{\beta}{2})$$

$$z_e = 0$$

$$[n_e] = \begin{bmatrix} \cos(\alpha+\beta) \\ \sin(\alpha+\beta) \\ 0 \end{bmatrix} \quad (3.9.2)$$

The surface of revolution is generated by rotation of the circular arc about the  $y_e$  axis and can be represented in coordinate system  $S_d$  as follows

$$[r_d] = [L_{de}][r_e] , \quad [n_d] = [L_{de}][n_e] \quad (3.9.3)$$

where (Fig. 9.2(b))

$$[L_{de}] = \begin{bmatrix} \cos\theta_p & 0 & \sin\theta_p \\ 0 & 1 & 0 \\ -\sin\theta_p & 0 & \cos\theta_p \end{bmatrix}$$

Then we obtain

$$x_d = [(\frac{d}{\cos\alpha} - \frac{b}{\cos\psi_n \cos\mu})\sin\alpha - 2\rho \sin \frac{\beta}{2} \sin(\alpha + \frac{\beta}{2})]\cos\theta_p$$

$$y_d = \frac{b}{\cos\psi_n \cos\mu} + 2\rho \sin \frac{\beta}{2} \cos(\alpha + \frac{\beta}{2})$$

$$z_d = [(\frac{d}{\cos\alpha} - \frac{b}{\cos\psi_n \cos\mu})\sin\alpha - 2\rho \sin \frac{\beta}{2} \sin(\alpha + \frac{\beta}{2})]\sin\theta_p$$

$$[n_d] = \begin{bmatrix} \cos\theta_p \cos(\alpha + \beta) \\ \sin(\alpha + \beta) \\ \sin\theta_p \cos(\alpha + \beta) \end{bmatrix} \quad (3.9.4)$$

The installment of the generating surface in coordinate system is the same as described in Section 3.3. The generated crowning pinion tooth surface can be obtained using the same approach as that described in section 3.4.

The meshing of gears using the crowning method described in this section has been simulated by numerical methods. The results of the investigation are illustrated with the following example.

Given: number of pinion teeth  $N_1 = 20$ , number of gear teeth  $N_2 = 40$ , diametral pitch in normal section  $P_n = 10 \text{ in}^{-1}$ , pressure angle in normal section  $\psi_n = 20^\circ$ , helical angle  $\beta = 15^\circ$ . The pinion tooth is crowned by involute surface with generatrix arc  $\rho = 30 \text{ in}$ . The involute surface is deviated from a cone (comparing  $\Sigma_p$  in figure 3.2(b) and (c)). The cone has half apex angle  $\alpha = 20^\circ$  and bottom radius  $R = 10.6 \text{ in}$ .

The topology of the pinion tooth surface provides a parabolic type of predesigned transmission errors with  $d = 6$  arc seconds (Fig. 2.3 (a)) and a path contact that is directed across the tooth surface (Fig. 3.1).

The influence of gear misalignment has been investigated with the developed computer program and the results of computation are represented in table 3.9.1 and 3.9.2 for crossed and intersected gear axes, respectively. The misalignment of gear axes is 5 arc minutes.

The results of computation show that the resulting function of transmission errors is a parabolic one. Thus the linear function of transmission errors that was caused by gear misalignment has been absorbed by the predesigned parabolic function. Fig. 3.14 show the results of transmissions errors for crossed helical gears with (Table 3.9.1) and without (Table 3.4.1) crowning of the pinion.

TABLE 3.9.1 - TRANSMISSION ERRORS OF CROSSED HELICAL GEARS

$\phi_1$ (deg)	-14	-11	-8	-5	-2	1	4
$\Delta\phi_2$ (sec)	-4.99	-1.51	0.65	1.51	1.05	-0.75	-3.87

TABLE 3.9.2 - TRANSMISSION ERRORS OF INTERSECTED HELICAL GEARS

$\phi_1$ (deg)	-11	-8	-5	-2	1	4	7
$\Delta\phi_2$ (sec)	-6.15	-2.72	-0.60	0.20	-0.32	-21.9	-5.40

#### 4. CROWNED HELICAL PINION WITH LONGITUDINAL PATH CONTACT

##### 4.1 Basic Concepts and Considerations

A longitudinal path of contact means that the gear tooth surfaces are in contact at a point at every instant and the instantaneous contact ellipse moves along but not across the surface (Fig. 4.1(a)). It can be expected that this type of contact provides improved conditions of lubrication. Until now

only the Novikov-Wildhaber's gears could provide a longitudinal path of contact. A disadvantage of this type of gearing is their sensitivity to the change of the center distance and the axes misalignment. The sensitivity to non-ideal orientation of the meshing gears cause a higher level of gear noise in comparison with regular involute helical gears. Litvin et al. (Litvin, 1985) proposed a compromising type of non-conformal helical gears that may be placed between regular helical gears and Novikov-Wildhaber helical gears. The gears of the proposed gear train are the combination of a regular involute helical gear and a specially crowned helical pinion. The investigation of transmission errors for helical gears with a longitudinal path of contact shows that their good bearing contact is accompanied with an undesirable increased level of linear transmission errors. The authors propose to compensate this disadvantage by a predesigned parabolic function of transmission errors, that will absorb the linear function of transmission errors (see section 2.3). The two following methods for derivation of the pinion tooth surface with the modified topology will now be considered.

#### 4.2 Method 1

Consider that two rigidly connected generating surfaces,  $\Sigma_G$  and  $\Sigma_p$ , are used for the generation of the gear and the pinion, respectively (Fig. 4.1(b)). Surface  $\Sigma_G$  is a plane and represents the surface of a regular rack-cutter; surface  $\Sigma_p$  is a cylindrical surface whose cross-section is a circular arc. We may imagine that while surfaces  $\Sigma_G$  and  $\Sigma_p$

translate, as the pinion and the gear rotate about their axes. To provide the predesigned parabolic function of transmission errors it is necessary to observe the following transmission functions by generation

$$\frac{V}{\omega_G} = r_2 = \text{const}, \quad \frac{V}{\omega_p} = r_2 \left( \frac{N_1}{N_2} - 2a\phi_p \right) = f(\phi_p) \quad (4.2.1)$$

Here:  $\omega_G$  and  $\omega_p$  are the angular velocities of pinion and gear by cutting;  $V$  is the velocity of the rack-cutter in translational motion;  $N_1$  and  $N_2$  are the gear and pinion tooth numbers;  $\phi_p$  is the angle of rotation of the pinion by cutting. The generated gears will be in point contact at every instant and transform rotation with the function

$$\phi_2(\phi_1) = \frac{N_1}{N_2} \phi_1 - a\phi_1^2 \quad 0 < \phi_1 < \frac{2\pi}{N_1} \quad (4.2.2)$$

This function relates the angles of rotation of the pinion and the gear,  $\phi_1$  and  $\phi_2$ , respectively, for one cycle of meshing. The predesigned function of transmission errors is

$$\Delta\phi_2 = -a\phi_1^2 \quad (4.2.3)$$

It is evident that after differentiation of function (4.2.2) we obtain that the gear ratio  $\omega_2/\omega_1$  satisfies equation (4.2.1), if  $\phi_1$  and  $\phi_2$  are used instead of  $\phi_p$  and  $\phi_G$ .

To apply this method of generation in practice it is necessary to vary the angular velocity of the pinion in the

process of its generation that may be accomplished by a computer controlled machine for cutting.

It is obvious that in this method the gear tooth surface is kept as regular skew involute surface as shown in Eq.(2.4.1) and (2.4.2) since its generating surface  $\Sigma_G$  is a plane and generating motion is  $V = \omega_G r_2$ . Now, let us derive the pinion tooth surface equations.

The pinion generating surface  $\Sigma_p$  is a cylindrical surface with circular arc as its cross section and can be represented in an auxiliary coordinate system  $S_a$  as follows: (see Fig. 4.2)

$$\tilde{r}_a = \begin{bmatrix} x_a \\ y_a \\ z_a \\ 1 \end{bmatrix} = \begin{bmatrix} R[\cos(\psi_n + \alpha) - \cos\psi_n] \\ R[\sin(\psi_n + \alpha) - \sin\psi_n] \\ t_p \\ 1 \end{bmatrix} \quad (4.2.4)$$

The unit surface normal is represented by

$$\tilde{n}_a = \begin{bmatrix} n_{ax} \\ n_{ay} \\ n_{az} \end{bmatrix} = \begin{bmatrix} \cos(\psi_n + \alpha) \\ \sin(\psi_n + \alpha) \\ 0 \end{bmatrix} \quad (4.2.5)$$

From coordinate system  $S_a$  to  $S_c$  the helical angle  $\beta_p$  is introduced. Based on Fig. 3.8(d) and  $[M_{ca}]$  as shown in Eq. (3.3.7), we have

$$\tilde{r}_c^{(p)} = \begin{bmatrix} x_c^{(p)} \\ y_c^{(p)} \\ z_c^{(p)} \\ 1 \end{bmatrix} = [M_{ca}] \tilde{r}_a = \begin{bmatrix} R[\cos(\psi_n + \alpha) - \cos\psi_n] - t_p \sin\beta_p \\ R[\sin(\psi_n + \alpha) - \sin\psi_n] \\ R[\cos(\psi_n + \alpha) - \cos\psi_n] + t_p \sin\beta_p \\ 1 \end{bmatrix} \quad (4.2.6)$$

$$\underline{n}_c^{(p)} = \begin{bmatrix} n_{cx}^{(p)} \\ n_{cy}^{(p)} \\ n_{cz}^{(p)} \end{bmatrix} = [L_{ca}] \underline{r}_a = \begin{bmatrix} \cos(\psi_n + \alpha) \cos \beta_p \\ \sin(\psi_n + \alpha) \\ \cos(\psi_n + \alpha) \sin \beta_p \end{bmatrix} \quad (4.2.7)$$

The generating process is described in the Fig. 3.5 where

$$S = r_2 \phi_p \left( \frac{N_1}{N_2} - a \phi_p \right) \quad (4.2.8)$$

The equation of meshing of the generating surface  $\Sigma_p$  and the helical pinion tooth surface is given by:

$$\underline{n}^{(p)} \cdot \underline{v}^{(p1)} = \underline{n}^{(p)} \cdot (\underline{v}^{(p)} - \underline{v}^{(1)}) = 0 \quad (4.2.9)$$

In the system shown in Fig. 3.5, it can be found that

$$\underline{v}^{(p)} = \begin{bmatrix} -\frac{ds}{dt} \\ 0 \\ 0 \end{bmatrix}$$

$$\underline{v}^{(1)} = \begin{bmatrix} -r_1 - y_c^{(p)} \\ -S + x_c^{(p)} \\ 0 \end{bmatrix} \omega_p \quad (4.2.10)$$

Taking into account that  $\frac{d\phi_p}{dt} = \omega_p$  and from Eq.(4.2.1) we have

$$\underline{v}^{(p1)} = \underline{v}^{(p)} - \underline{v}^{(1)} = \omega_p \begin{bmatrix} -r_2 \left( \frac{N_1}{N_2} - 2a \phi_p \right) + r_1 + y_c^{(p)} \\ -x_c^{(p)} + r_2 \phi_p \left( \frac{N_1}{N_2} - a \phi_p \right) \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} y_c^{(p)} + 2r_2 a \phi_p \\ -x_c^{(p)} + r_2 \phi_p \left( \frac{N_1}{N_2} - a \phi_p \right) \\ 0 \end{bmatrix} \omega_p \quad (4.2.11)$$

Substituting Eq.(4.2.6), (4.2.7) and (4.2.11) into Eq. (4.2.9).  
the equation of meshing can be obtained as:

$$\begin{aligned} & \cos(\psi_n + \alpha) \cos \beta_p \{ R[\sin(\psi_n + \alpha) - \sin \psi_n] + 2r_2 a \phi_p \} \\ & + \sin(\psi_n + \alpha) \{ -R[\cos(\psi_n + \alpha) - \cos \psi_n] - t \sin \beta_p \} + r_2 \phi_p \left( \frac{N_1}{N_2} - a \phi_p \right) \\ & = 0 \end{aligned} \quad (4.2.12)$$

Equation of meshing gives the relation among three variables,  
that is,  $t_p$ ,  $\alpha$  and  $\phi_p$ . It is obvious that from Eq. 4.2.12, the  
equation of meshing can be rewritten as:

$$\begin{aligned} t_p &= \cotan(\psi_n + \alpha) \cotan(\beta_p) \{ R[\sin(\psi_n + \alpha) - \sin \psi_n] + 2r_2 a \phi_p \} \\ &+ \frac{1}{\sin \beta_p} \{ -R[\cos(\psi_n + \alpha) - \cos \psi_n] + r_2 \phi_p \left( \frac{N_1}{N_2} - a \phi_p \right) \} \end{aligned} \quad (4.2.13)$$

The pinion tooth surface can be determined with the same approach

as that described in the section 3.4 and represented by Eq.(3.4.8) and (3.4.9) together with Eq.(4.2.6),(4.2.7) and (4.2.13).

#### 4.3 Method 2

The derivation of the crowned pinion tooth surface is based on two stages of synthesis. On the first stage it is assumed the pinion tooth surface  $\Sigma_1$  is exact conjugate surface to the gear tooth surface which is regular skew involute surface under the condition that the rotation transformed by the pinion and the gear is described in Eq.(4.2.2) with predesigned parabolic function of transmission errors.

On the second stage of synthesis it is necessary to localize the bearing contact and substitute the instantaneous line contact by a point contact. This becomes possible if the pinion tooth surface will be deviated as it is shown in Fig. 4.1(c). Only a narrow strip,  $L$ , will be kept while  $\Sigma_1$  will be changed into  $\Sigma_1^1$ . The deviation of  $\Sigma_1^1$  with respect to  $\Sigma_1$  may be accomplished in various ways, for instance, in such a way, that the cross-section of  $\Sigma_1^1$  is only a circular arc. The generation of  $\Sigma_1^1$  requires a computer controlled machine to relate the motions of the tool surface and the being generated pinion surface  $\Sigma_1^1$ . The tool surface (it may be only a plane) and  $\Sigma_1^1$  will be in point contact in the process of generation.

Now, let us derive the pinion tooth surface equations based on the two stages discussed above. First, we can derive the intermediate pinion tooth surface  $\Sigma_1$  as that generated by gear

tooth surface  $\Sigma_1$ . The relation of rotation for the pinion and the gear is shown in Eq.(4.2.2). The gear tooth surface  $\Sigma_2$  and its unit normal are shown in Eq.(2.4.3) and (2.4.4).

To represent the gear tooth surface  $\Sigma_2$  in the fixed coordinate system  $S_f$  in Fig. 4.3, it is necessary to transform  $[r_2]$  and  $[n_2]$  into  $[r_f^{(2)}]$  and  $[n_f^{(2)}]$  as:

$$[r_f^{(2)}] = [M_{f2}][r_2]$$

$$[n_f^{(2)}] = [L_{f2}][n_2] \quad (4.3.1)$$

$$\text{where } [M_{f2}] = \begin{bmatrix} -\cos\phi_2 & \sin\phi_2 & 0 & 0 \\ -\sin\phi_2 & -\cos\phi_2 & 0 & 0 \\ 0 & 0 & 1 & C \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Substituting Eq.(2.4.3) and (2.4.4) into Eq.(4.3.1) we obtain:

$$[r_f^{(2)}] = \begin{bmatrix} x_f^{(2)} \\ y_f^{(2)} \\ z_f^{(2)} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\cos^2\psi_n}{\cos\psi_c} \cos(\phi_G - \psi_c - \phi_2) [-t_G \sin\beta_p + r_2\phi_G] - r_2 \sin(\phi_G - \phi_2) \\ C - \frac{\cos^2\psi_n}{\cos\psi_c} \cos(\phi_G - \psi_c - \phi_2) [-t_G \sin\beta_p + r_2\phi_G] - r_2 \sin(\phi_G - \phi_2) \\ t_G^2 \cos^2\beta_p (1 + \sin\psi_n + g^2\beta_p) - r_2\phi_G \sin\psi_n \operatorname{tg}\beta_p \\ 1 \end{bmatrix} \quad (4.3.2)$$

$$[n_f^{(2)}] = \begin{bmatrix} n_{fx}^{(2)} \\ n_{fy}^{(2)} \\ n_{fz}^{(2)} \end{bmatrix} = \begin{bmatrix} \cos\psi_n \cos\beta_p \cos(\phi_G - \phi_2) + \sin\psi_n \sin(\phi_G - \phi_2) \\ -\cos\psi_n \cos\beta_p \sin(\phi_G - \phi_2) + \sin\psi_n \cos(\phi_G - \phi_2) \\ \cos\psi_n \sin\beta_p \end{bmatrix} \quad (4.3.3)$$

The generating process is shown in Fig. 4.3 where  $\phi_1$  and  $\phi_2$  is given by Eq. (4.2.2). The equation of meshing of the generating surface  $\Sigma_2$  and generated surface  $\Sigma_1$  is described as:

$$\tilde{N}^{(2)} \cdot \tilde{V}^{(21)} = \tilde{N}^{(2)} (\tilde{V}^{(2)} - \tilde{V}^{(1)}) = 0 \quad (4.3.4)$$

The relative velocity  $\tilde{V}^{(21)}$  can be expressed as

$$\tilde{V}^{(21)} = \tilde{V}^{(2)} - \tilde{V}^{(1)} = \omega_2 \times (\overline{O_2 O_1} + \tilde{r}_f) - \omega_1 \times \tilde{r}_f \quad (4.3.5)$$

where

$$\omega^{(1)} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \omega_1; \quad \omega^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_1 \left[ \frac{N_1}{N_2} - 2a\phi_1 \right]$$

$$\overline{O_2 O_1} = \begin{bmatrix} 0 \\ -C \\ 0 \end{bmatrix}; \quad \tilde{r}_f = \begin{bmatrix} x_f^{(2)} \\ y_f^{(2)} \\ z_f^{(2)} \end{bmatrix};$$

and the relation between  $\omega_1$  and  $\omega_2$  is obtained by taking derivative of Eq. (4.2.2). By rearranging and simplifying Eq. (4.3.5), we obtain

$$\tilde{V}^{(21)} = \begin{bmatrix} -y_f^{(2)} \\ x_f^{(2)} \\ 0 \end{bmatrix} \omega_1 \left[ \frac{N_2}{N_1} + 1 - 2a\phi_1 \right] + \begin{bmatrix} C \\ 0 \\ 0 \end{bmatrix} \omega_1 \left[ \frac{N_1}{N_2} - 2a\phi_1 \right] \quad (4.3.6)$$

Substituting Eq.(4.3.6) (4.3.2) and (4.3.3) into Eq.(4.3.4) and

simplifying it, we can obtain:

$$\cos(\phi_G - \phi_2 - \psi_c) = \cos\psi_c \left[ 1 - \frac{2a\phi_1}{N_1/N_2} \right] \quad (4.3.7)$$

where  $\psi_c$  is the pressure angle in transverse section of helical gear. Equation 4.3.7 is the equation of meshing which describes the relation between  $\phi_G$  and  $\phi_2$ . We can rearrange Eq. 4.3.7 as:

$$\phi_G = \phi_2 + \lambda(\phi_1) = \frac{N_1}{N_2} \phi_1 - a\phi_1^2 + \lambda(\phi_1) \quad (4.3.8)$$

where  $\lambda(\phi_1) = \psi_c - \arccos \cos\psi_c \left[ 1 - \frac{2a\phi_1}{1 + N_1/N_2} \right]$

The pinion tooth surface  $\Sigma_1$  and its normal can be determined by transforming  $\tilde{r}_f^{(2)}$  into coordinate system  $S_1$  as

$$[r_1] = [M_{1f}] [\tilde{r}_f^{(2)}] \quad (4.3.9)$$

$$[n_1] = [L_{1f}] [n_f^{(2)}]$$

$$\text{where } [M_{1f}] = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 & 0 & 0 \\ \sin\phi_1 & \cos\phi_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Substituting (4.3.2) (4.3.3) into Eq.(4.3.9), we can get

$$[r_1] = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \quad (4.3.10)$$

$$[n_1] = \begin{bmatrix} n_{1x} \\ n_{1y} \\ n_{1z} \end{bmatrix}$$

where

$$x_1 = \frac{\cos^2 \psi_n}{\cos \psi_c} \cos(\phi_1 + \psi_c - \lambda(\phi_1)) [-t_G \sin \beta_p + r_2 \phi_G] + r_2 \sin(\phi_1 - \lambda(\phi_1)) - C \sin \phi_1$$

$$y_1 = \frac{\cos^2 \psi_n}{\cos \psi_c} \sin(\phi_1 + \psi_c - \lambda(\phi_1)) [-t_G \sin \beta_p + r_2 \phi_G] - r_2 \cos(\phi_1 - \lambda(\phi_1)) + C \cos \phi_1$$

$$z_1 = t_G \cos \beta_p (1 + \sin^2 \psi_n + g^2 \beta_p) - r_2 \phi_G \sin^2 \psi_n t_g \beta_p$$

$$n_{1x} = \frac{\cos \psi_n \cos \beta_p}{\cos \psi_c} \cos(\phi_1 - \lambda(\phi_1) + \psi_c)$$

$$n_{1y} = \frac{\cos \psi_n \cos \beta_p}{\cos \psi_c} \sin(\phi_1 - \lambda(\phi_1) + \psi_c)$$

$$n_{1z} = \cos \psi_n \sin \beta_p$$

Equations (4.3.10), (4.3.8) and (4.2.2) represent pinion tooth surface  $\Sigma_1$  in coordinate system  $S_1$ . But  $\Sigma_1$  is only an intermediate surface. Now we come to the second stage, that is, to deviate  $\Sigma_1$  into  $\Sigma_1^1$ . The surface  $\Sigma_1^1$  must satisfy such condition that when it is in mesh with gear tooth surface  $\Sigma_2$  without misalignment, the contact path will be in longitudinal direction. The surface  $\Sigma_1^1$  will be formed in two steps: (i) the contact path curve in  $\Sigma_1$  is chosen and kept. (ii) the profile of surface  $\Sigma_1$  in transverse section is replaced

by a smaller circular arc attached on the L in such a way that the original normal of  $\Sigma_1$  along L is kept.

To choose curve L on  $\Sigma_1$  we could define that the contact path on the gear tooth surface is in the middle of the tooth, that is

$$r = r_2 \quad (4.3.11)$$

where  $r = \sqrt{x_2^2 + y_2^2}$  is the distance of the surface point to gear axis in transverse section. Substituting Eq. (2.4.3) into Eq. (4.3.10) and simplifying it, we can obtain the relation of surface parameter along the contact path as

$$r_2 \phi_G - t_G \sin \beta_p = 0 \quad (4.3.12)$$

Applying Eq.(4.3.12) to Eq.(4.3.10), we can find the curve L on pinion tooth surface  $\Sigma_1$  as:

$$\begin{aligned} x_1 &= r_2 \sin(\phi_1 - \lambda(\phi_1)) - C \sin \phi_1 \\ y_1 &= -r_2 \cos(\phi_1 - \lambda(\phi_1)) + C \cos \phi_1 \\ z_1 &= r_2 \operatorname{ctg} \beta_p \left( \frac{N_1}{N_2} \phi_1 - a \phi_1^2 + \lambda(\phi_1) \right) \end{aligned} \quad (4.3.13)$$

where the  $\phi_1$  is the curve parameter and  $\lambda(\phi_1)$  is described in Eq. (4.3.8).

Now we should attach the circular arc to the L described by Eq. (4.3.13) in the transverse section to form new surface  $\Sigma_1^1$ . Also the tangent of the arc at the point on the line L must be perpendicular to the normal of  $\Sigma_1$  described in Eq. (4.3.10). As shown in Fig. 4.4, the equation of new surface  $\Sigma_1^1$  are:

$$x_1' = x_1 + r[\cos(\mu+\alpha) - \cos\mu]$$

$$y_1' = y_1 + R[\sin(\mu+\alpha) - \sin\mu] \quad (4.3.14)$$

$$z_1' = z_1$$

where  $\mu = \phi_1 - \lambda(\phi_1) + \psi_c$ . Equation (4.3.14) describes the new pinion surface  $\Sigma_1^1$ . In Eq. (4.3.14) when  $\alpha = 0$ , the designed contact path L is obtained.

#### 4.4 Discussion and Example

In section 4.2 and 4.3, two methods have been presented with derivation of the equations of pinion tooth surface. Comparing the two methods for the generation of the pinion tooth surface, it may be concluded that both provide a localized bearing contact, a longitudinal path of contact and predesigned parabolic function of transmission errors. The difference between these methods is that the tool and pinion tooth surfaces are in line contact by applying the first method for generation and in point contact by the second one. The disadvantage of both methods for crowning of the pinion is that the transmission errors caused by

gear misalignment are large and it is necessary to envision a high level of the predesigned parabolic function for the absorption of transmission errors. This is illustrated with the following example (the algorithms for simulation have been discussed in Section 3.8).

Given (the data is from ref. 3): pinion tooth number  $N_1 = 12$ , gear tooth number  $N_2 = 94$ ; diametral pitch in normal section  $P_n = 2 \text{ in}^{-1}$ ; pressure angle in normal section  $\psi_n = 30^\circ$ ; helical angle  $\beta = 15^\circ$ .

The pinion tooth surface is a crowned surface whose cross-section is an arc of a circle of the radius 0.3584in. The predesigned parabolic function is of the level  $d = 25$  arc seconds (Fig. 2.3(a)).

Consider now that the axes of the gear and the pinion are crossed and the crossing angle is 3 arc minutes. The computer program for the simulation of meshing provides the data of transmission errors that is given in Table 4.1. The data of Table 4.1 shows that the resulting function of transmission errors is a parabolic function. Thus, the linear function of transmission errors caused by misalignment of gear axes has been absorbed by the predesigned parabolic function.

Table 4.2 represents the transmission errors for the same helical gears for the case when the gear axes are intersected and form an angle of 3 arc minutes. The resulting function of transmission errors is again a parabolic function with the level  $d = 26.2$  arc seconds. The relatively high level of transmission errors is the price that must be paid for the longitudinal path

of contact. However, the proposed topology of the pinion tooth surface provides a reduction of the level of gear noise since the linear function of transmission errors is substituted by a parabolic function.

TABLE 4.1 TRANSMISSION ERRORS FOR CROSSED HELICAL GEARS

$\phi_1$ (deg)	-23	-18	-13	-8	-3	2	7
$\Delta\phi_2$ (sec)	-17.94	-3.06	5.64	8.23	4.84	-4.39	-19.37

TABLE 4.2 TRANSMISSION ERRORS OF INTERSECTED HELICAL GEARS

$\phi_1$ (deg)	-20	-15	-10	-5	0	5	10
$\Delta\phi_2$ (sec)	-23.06	-8.50	0.15	2.96	0.00	-8.66	-22.95

## 5. DEFORMATION OF HELICAL GEAR SHAFT

### 5.1 Basic Concepts and Considerations

Deformation of gear shafts always exists when the gears are used to transmit power. This is because under the load the force applied on gear tooth surface is transferred to the gear shaft and the shaft is not rigid body. It can be proven that the deformation of gear shaft results in the same effects as misalignment induced by assembly. The misalignment from assembly could be reduced to as little as possible. But the shaft deformation is inevitable. Fortunately, all the transmission

errors and shift of bearing contact due to deformation of gear shaft can be compensated by crowning helical pinion tooth surface with the methods discussed in Chapters 3 and 4.

## 5.2 Force Applied on Gear Shaft

When the pinion and gear mesh a force, between their tooth surfaces, is applied on the contact point. The direction of the force is along the normal of the contact point. For the case of regular helical gears, in the fixed coordinate system  $S_f$  as described in section 3.8, the force direction is a constant since the normal of the contact point is a constant (Litvin, 1968) and can be expressed as:

$$\vec{n}_c = \begin{bmatrix} \cos\psi_n \cos\beta_p \\ \sin\psi_n \\ \cos\psi_n \sin\beta_p \end{bmatrix} \quad (5.2.1)$$

As shown in Fig. 5.1, the force applied on the pinion tooth surface is:

$$\vec{F} = \vec{F}_t + \vec{F}_z = -F \begin{bmatrix} \cos\psi_n \cos\beta_p \\ \sin\psi_n \\ \cos\psi_n \sin\beta_p \end{bmatrix} \quad (5.2.2)$$

$$\text{where } \vec{F}_t = -F \begin{bmatrix} \cos\psi_n \cos\beta_p \\ \sin\psi_n \\ 0 \end{bmatrix}; \quad \vec{F}_z = -F \begin{bmatrix} 0 \\ 0 \\ \cos\psi_n \sin\beta_p \end{bmatrix}$$

$F_t$  is called as transverse force since it is applied on the transverse section of the pinion and the gear and  $F_z$  is called as axial force. Transverse force  $F_t$  is always in tangency with the base circle in transverse section. Therefore, if the torque transferred by the gears is constant, the magnitude of transverse force is also constant. So is the magnitude of axial force since it has certain ratio with transverse force.

Both transverse force and axial force can be transferred to the axis of gear shaft with resultant torque (designated by  $F_t^*$  and  $F_z^*$ ). For the transverse force, the resultant torque is the torque transferred by the gear. For the axial force, the resultant torque is balanced by the support bearings. Assuming the force is applied on the middle section of the gear, after the force is decomposed and transferred to the axis of gear shaft as shown in Fig. 5.1 (b), both the transverse force and axial force will act on the origin of coordinate system  $S_f$ . Actually, both forces will cause deformation of the shaft. But since the axial force only cause very small tension or compression of the shaft, it can be neglected.

### 5.3 Modelling of Shaft Deformation

As shown in Fig. 5.2, the transverse force  $F_t$  is applied on the point A which is the center of the shaft corresponding to the pinion or gear middle cross section. Under the force, the deformation of the shaft is composed of two parts, that is, deflection of shaft at the point A designated by  $V_p$  and rotation of the shaft cross section with A as a center designated by

$\lambda_p$ . The value  $V_p$  and  $\lambda_p$  depend on the magnitude of transverse force, geometry and material of shaft and the way how the shaft is supported. (Timoshenko 1973).

Now, let us model the deformation of the helical pinion shaft. For convenience and consistency with the previous chapters, We establish our coordinate system as follows:

(1) As shown in Fig. 5.3a  $S_f$  is a fixed coordinate system and rigidly connected to the frame.  $S_a$  is an auxiliary coordinate system with the  $Y_a$  axis coincident with the direction of transverse force. The angle  $\psi_c$  is the pressure angle in the transverse section of helical gear. The matrix  $[M_{fa}]$  can transfer vector from  $S_a$  to  $S_f$  and is expressed as:

$$[M_{fa}] = \begin{bmatrix} \sin\psi_c & \cos\psi_c & 0 & 0 \\ -\cos\psi_c & \sin\psi_c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.1)$$

(2) Coordinate system  $S_{f1}$  and  $S_a$ , [Fig. 5.3(b)] are connected to the axis of pinion shaft. Without deformation,  $S_f'$  and  $S_a'$  are coincident with  $S_f$  and  $S_a$  respectively. The matrix  $[M_{a'f'}]$  can transfer vector from  $S_f'$  to  $S_a'$ , and is expressed as:

$$[M_{a'f'}] = \begin{bmatrix} \sin\psi_c & -\cos\psi_c & 0 & 0 \\ \cos\psi_c & \sin\psi_c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.2)$$

(3) Coordinate systems  $S_a$  and  $S_a'$  are not coincident when shaft deformation occurs. As shown in Fig. 5.2, their relation can be

expressed as:

$$M_{aa'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\lambda_p & -\sin\lambda_p & -v_p \\ 0 & \sin\lambda_p & \cos\lambda_p & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.3)$$

(4) As described in previous chapter, pinion tooth surface can be expressed in coordinate systems  $S_1$  as a position vector  $\underline{r}_1$  with its normal  $\underline{n}_1$ . The relation of  $S_1$  and  $S_f'$  is shown in Fig. 5.3(c) and expressed as:

$$M_{f,1} = \begin{bmatrix} \cos\phi_1 & \sin\phi_1 & 0 & 0 \\ -\sin\phi_1 & \cos\phi_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.4)$$

After the coordinate systems are established, it is easy to find that the deformation of the pinion shaft can be modelled by matrix  $m_{ff'}$  as:

$$[M_{ff'}] = [M_{fa}] [M_{aa'}] [M_{a',f'}]$$

$$= \begin{bmatrix} 1 + \cos^2\psi_c (\cos\lambda_p - 1) & , \sin\psi_c \cos\psi_c (\cos\lambda_p - 1) & , -\cos\psi_c \sin\lambda_p & , -v_p \cos\psi_c \\ \sin\psi_c \cos\psi_c (\cos\lambda_p - 1) & , 1 + \sin^2\psi_c (\cos\lambda_p - 1) & , -\sin\psi_c \sin\lambda_p & , -v_p \sin\psi_c \\ \sin\lambda_p \cos\psi_c & , \sin\lambda_p \sin\psi_c & , \cos\lambda & , 0 \\ 0 & , 0 & , 0 & , 1 \end{bmatrix} \quad (5.3.5)$$

Since  $\lambda_p$  is very small angle, it is reasonable to use  $\lambda_p$  instead of  $\sin\lambda_p$  and one instead of  $\cos\lambda_p$ . Therefore, Eq. (5.3.5) can be written as:

$$[M_{ff'}] = \begin{bmatrix} 1 & 0 & -\lambda_p \cos\psi_c & -V_p \cos\psi_c \\ 0 & 1 & -\lambda_p \sin\psi_c & -V_p \sin\psi_c \\ \lambda_p \cos\psi_c & \lambda_p \sin\psi_c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.6)$$

Also, for the gear shaft, the coordinate systems  $S_G$ ,  $S_{G'}$ ,  $S_b$ , and  $S_b'$  are used instead of  $S_f$ ,  $S_{f'}$ ,  $S_a$  and  $S_{a'}$ , and  $\lambda_G$ , and  $V_G$  are used instead of  $\lambda_p$  and  $V_p$ . Then, using the same ideas, the deformation of gear shaft can be modelled by matrix  $[m_{aa'}]$  as:

$$M_{GG'} = \begin{bmatrix} 1 + \cos^2\psi_c (\cos\lambda_G - 1) & , & \sin\psi_c \cos\psi_c (\cos\lambda_G - 1) & , & \cos\psi_c \sin\lambda_G & , & V_G \cos\psi_c \\ \sin\psi_c \cos\psi_c (\cos\lambda_G - 1) & , & 1 + \sin^2\psi_c (\cos\lambda_G - 1) & , & \cos\psi_c \sin\lambda_G & , & V_G \cos\psi_c \\ -\sin\lambda_G \cos\psi_c & , & -\sin\lambda_G \sin\psi_c & , & \cos\lambda_G & , & 0 \\ 0 & , & 0 & , & 0 & , & 1 \end{bmatrix} \quad (5.3.7)$$

Or considering that  $\lambda_G$  is very small angle:

$$[M_{GG'}] = \begin{bmatrix} 1 & 0 & +\lambda_G \cos\psi_c & V_G \cos\psi_c \\ 0 & 1 & +\lambda_G \sin\psi_c & V_G \sin\psi_c \\ -\lambda_G \cos\psi_c & -\lambda_G \sin\psi_c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.8)$$

It must be emphasized that the gear tooth surfaces are expressed

in coordinate system  $S_2$ . The relation of  $S_2$  and  $S_f'$  is shown in Fig. 5.4 and expressed as:

$$[M_{G',2}] = \begin{bmatrix} -\cos\phi_2 & \sin\phi_2 & 0 & 0 \\ -\sin\phi_2 & -\cos\phi_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.9)$$

Also as shown in Fig. 5.5, the coordinate system  $S_f$  and  $S_G$  are not coincident. There is a center distance  $C$  between their origins. Therefore  $M_{fG}$  is represented as:

$$[M_{fG}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & C \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.10)$$

Now it is easy to simulate the performance of the gears with the deformation of their shafts. Assuming, there is a helical pinion in coordinate system  $S_1$  represented by position vector  $\underline{r}_1$  and normal vector  $\underline{n}_1$  and a helical gear in coordinate system  $S_2$  represented by  $\underline{r}_2$  and  $\underline{n}_2$ , we can write the tooth contact equation as:

$$[M_{ff,}] [r_{f,1}] [\underline{r}_1] = [M_{fG}] [M_{GG,}] [M_{G',2}] [\underline{r}_2] \quad (5.3.11)$$

$$[L_{ff,}] [L_{f,1}] [\underline{r}_1] = [L_{fG}] [L_{GG,}] [L_{G',2}] [\underline{n}_2]$$

where  $L$  matrices are 3 x 3 matrices from corresponding 4 x 4  $M$  matrices, crossing 4th row and 4th column and  $M$  matrices can be found in this section. Applying the same ideas discussed in Chapter 2 and used in section 3.8, we can find the transmission errors and the shift of the bearing contact by computer aided

simulation.

It is interesting to mention that the deformation of the shaft can be expressed as a combination of misalignment with crossing axes, misalignment with intersecting axes, and change of the center distance. This is because, for example

$$\begin{aligned}
 [M_{ff}] &= \begin{bmatrix} 1 & 0 & -\lambda_p \cos \psi_c & -V_p \cos \psi_c \\ 0 & 1 & -\lambda_p \sin \psi_c & -V_p \sin \psi_c \\ \lambda_p \cos \psi_c & \lambda_p \sin \psi_c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & -\lambda_p \cos \psi_c & 0 \\ 0 & 1 & 0 & 0 \\ \lambda_p \cos \psi_c & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\lambda_p \sin \psi_c & 0 \\ 0 & \lambda_p \sin \psi_c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & -V_p \cos \psi_c \\ 0 & 1 & 0 & -V_p \sin \psi_c \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3.12)
 \end{aligned}$$

where the three decomposed matrices are represented by the matrix for crossing axis with small angle  $\lambda_p \cos \psi_c$ , matrix for intersecting axis with small angle  $\lambda_p \sin \psi_c$ , and matrix for axis displacement.

#### 5.4 Example and Discussion

The results of investigation in this chapter are illustrated with the following example. Given: number of pinion tooth  $N_1 = 20$ , number of gear tooth  $N_2 = 40$ , diametral pitch in normal section  $P_n = 10 \text{ in}^{-1}$ , pressure angle in normal

section  $\psi_n = 20^\circ$ , helical angle  $\beta = 15^\circ$ , gear tooth length  $L = 5/P_n$ . Also assume deformation values  $V_p = V_G = 0.0125$  in,  $\lambda_p = \lambda_G = 2$  min. The computer program for simulation provides the data of transmission errors in Table 5.1

From Table 5.1, it is known that the transmission errors due to gear shaft deformation is an approximately linear function for regular skew involute pinion and gear. Similar to the case of gear axis misalignment, the linear function of transmission errors can be absorbed by predesigned parabolic function of transmission errors that is obtained by crowning helical pinion tooth surface.

TABLE 5.1 TRANSMISSION ERRORS FOR HELICAL GEAR WITH DEFORMED SHAFT

$\phi_1$ (deg)	3	6	9	12	15	8	21
$\Delta\phi_2$ (sec)	1.38	3.16	4.74	6.32	7.90	9.48	11.06

## 6. CONCLUSION

Several methods of crowning helical pinion tooth surface have been developed. The modified pinion tooth surface can provide predesigned parabolic function of transmission errors that are able to absorb linear function of transmission errors induced by misalignment. Also, the modified pinion tooth surface can improve the bearing contact. Principles of computer aided simulation of meshing, contact, and respective computer programs have also been developed. The numerical results of examples of

crowned helical pinion in mesh with regular helical gear show that the ideas of crowning are useful to get favourable bearing contact and allowable transmission errors. But the synthesis of pinion tooth surface should be based on a compromise between the requirements of transmission errors and the patterns of the bearing contact.

## REFERENCES

1. Chironis, N.P., Design of Novikov Gear, in Gear Design and Application, N.P. Chironis (ed.), McGraw-Hill, New York.
2. Litvin, F.L., 1968, Theory of Gear, 2nd edition, Moscow (in Russian). The English version is in press, NASA publications.
3. Litvin, F.L. and Tsay, C., 1985, Helical Gears with Circular Arc Teeth: Simulation of Conditions of Meshing and Bearing Contact, Journal of Mechanisms, Transmissions and Automation in Design, Vol. 107, No. 4, pp. 556-564.
4. Litvin, F.L. and Zhang, J., et al., 1987, Crowned Spur Gears: Optimal Geometry and Generation, AGMA paper, 87FTM7.
5. Maag Information 18, Topological Modification, Zurich.
6. Wildhaber, E., 1962, Method and Machine for Producing Crowned Teeth, U.S.A. Patent 3,046,844.
7. Spur Gears: Optimal Geometry, Methods for Generation and Tooth Contact Analysis (TCA) Program, NASA Contractor Report, 4135, 1988.
8. S.P. Timoshenko and J.M. Gere, Mechanics of Materials, 1973, Van Nostrand Reinhold Company Ltd.

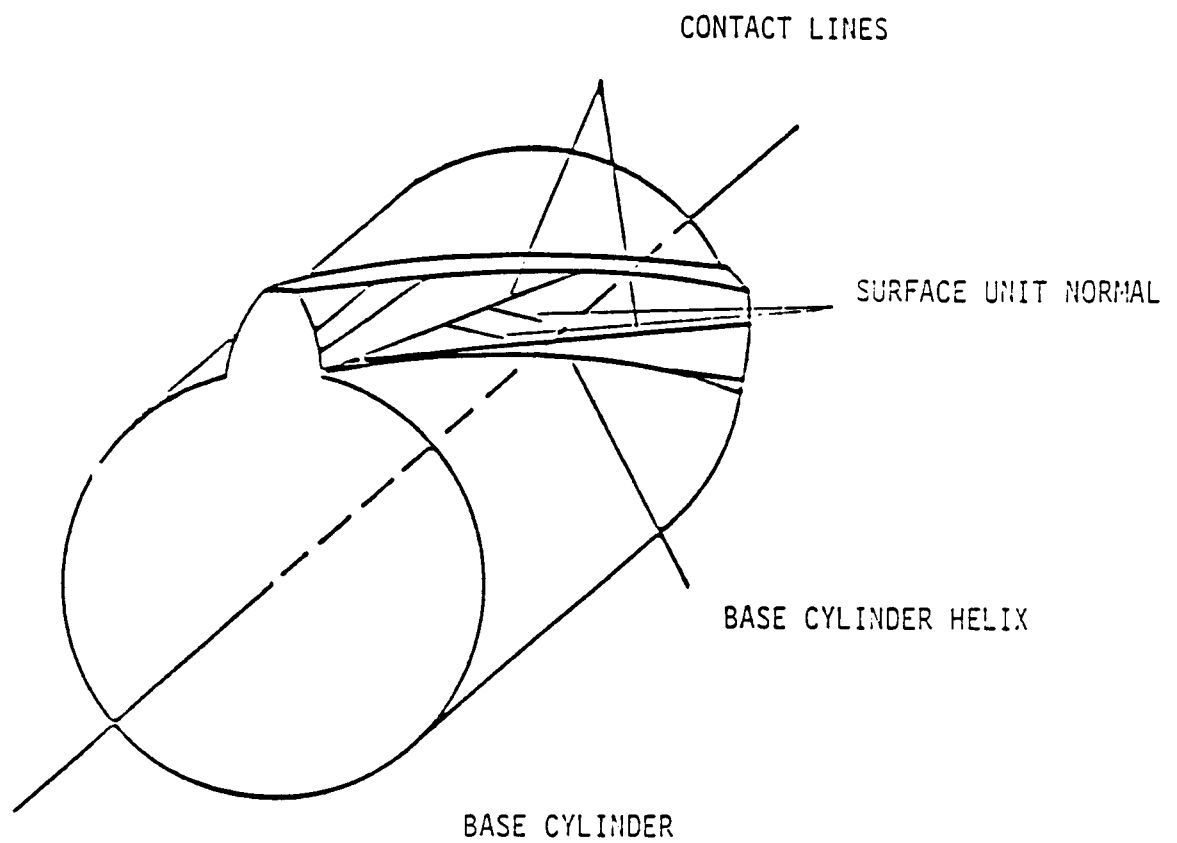


FIG. 1.1 Screw Involute Helical Gear

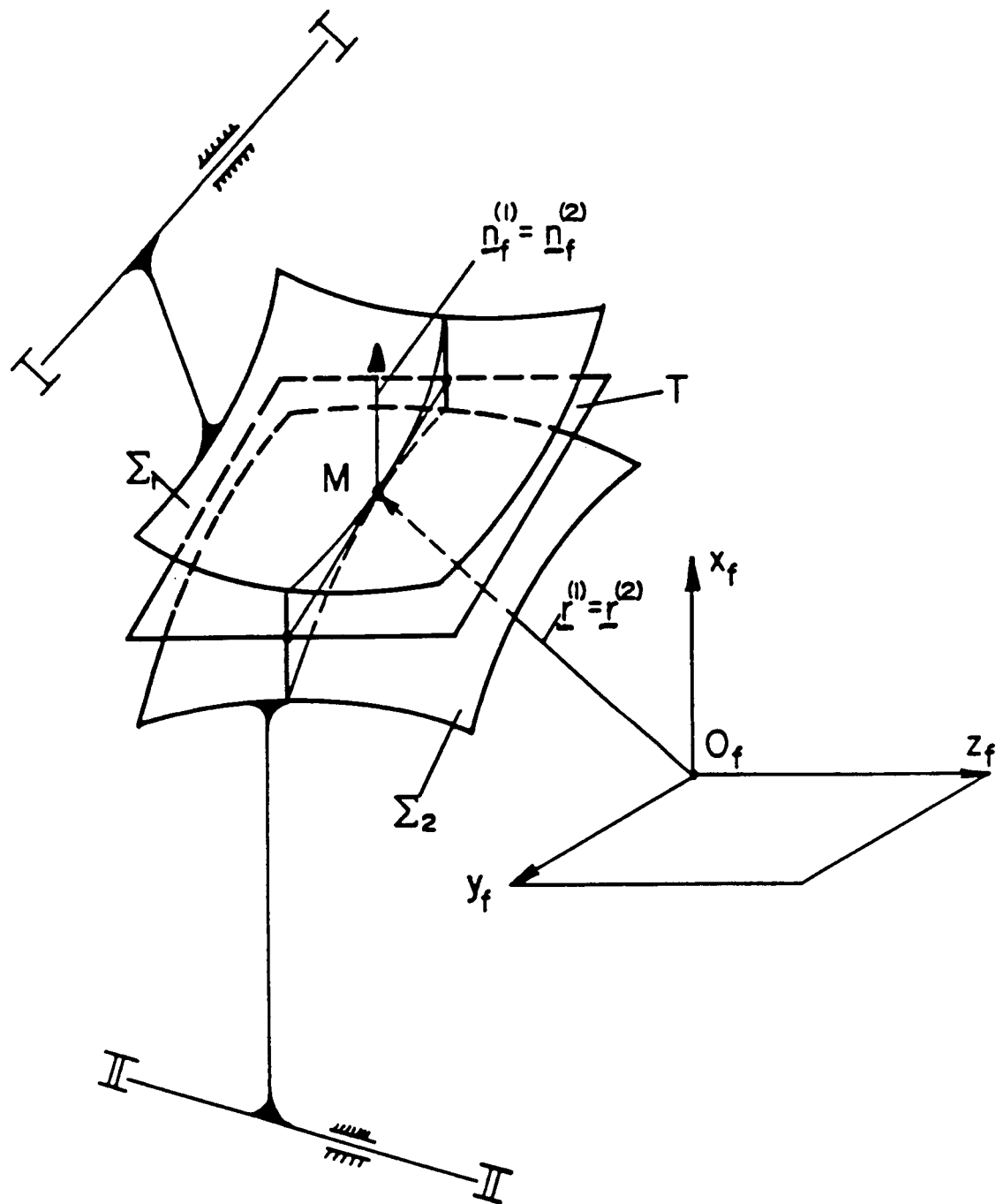


FIG. 2.1 Contacting Tooth Surfaces

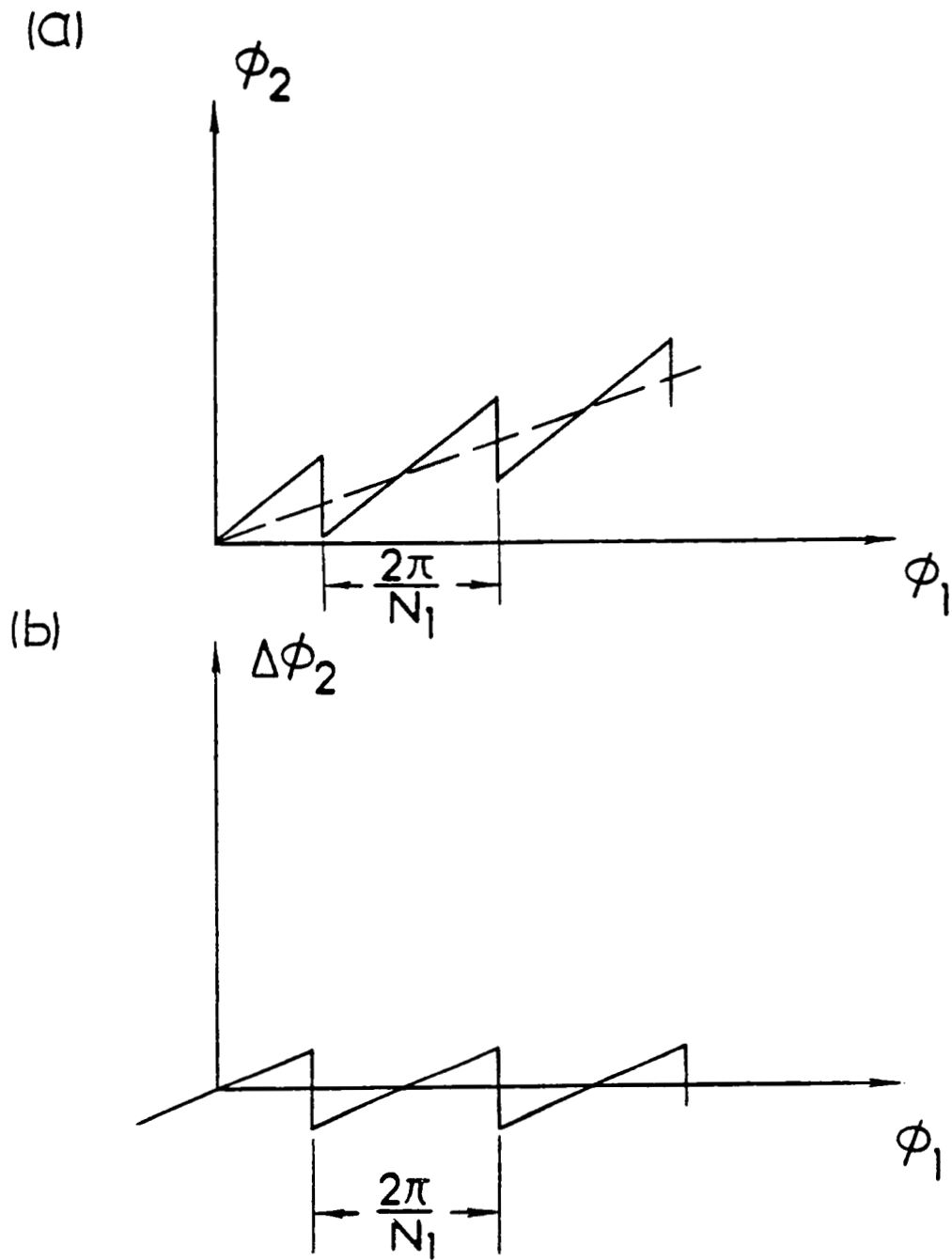


FIG. 2.2 Transmission Error Caused by Gear Misalignment

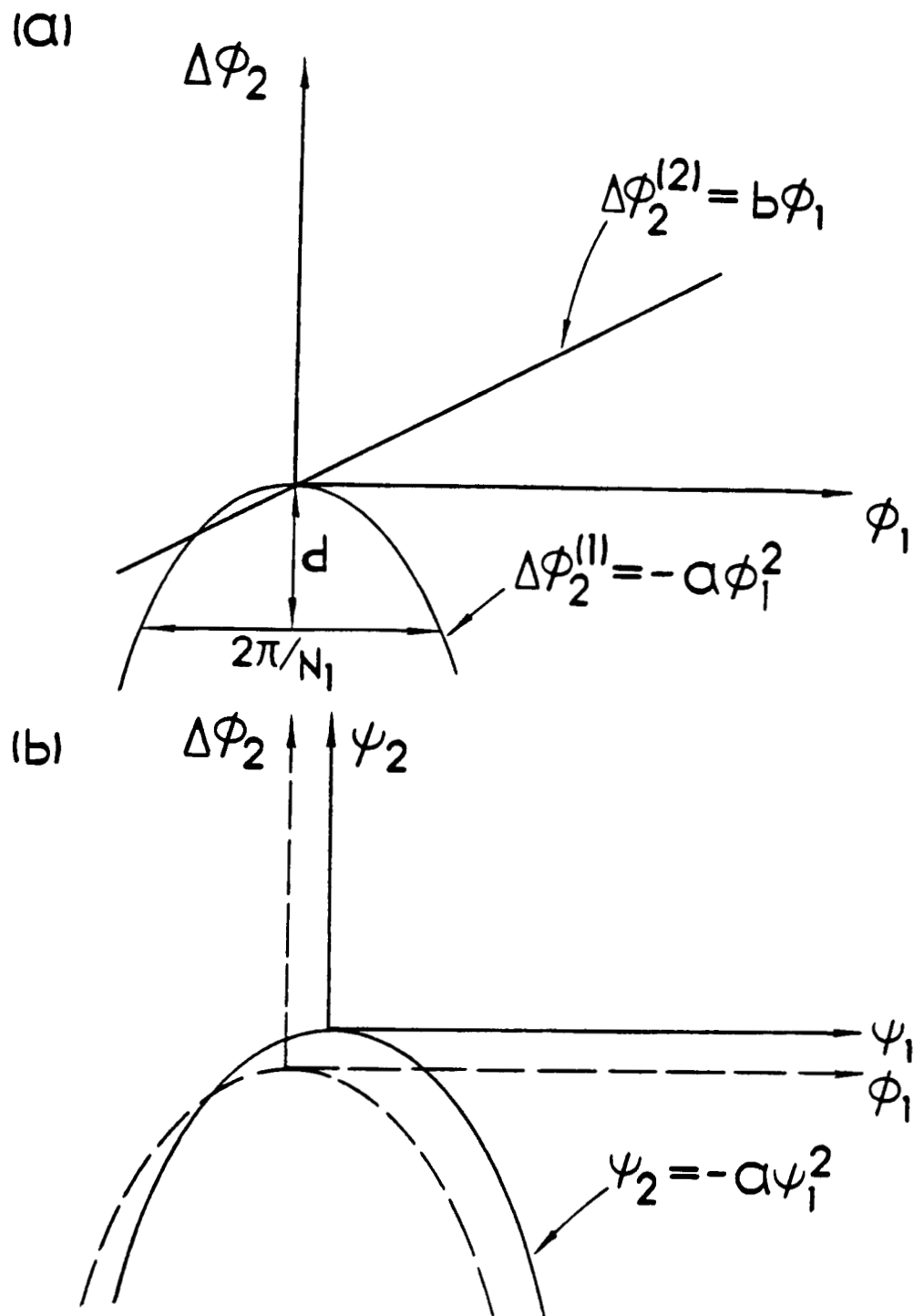


FIG . 2.3 Interaction of Parabolic and Linear Functions

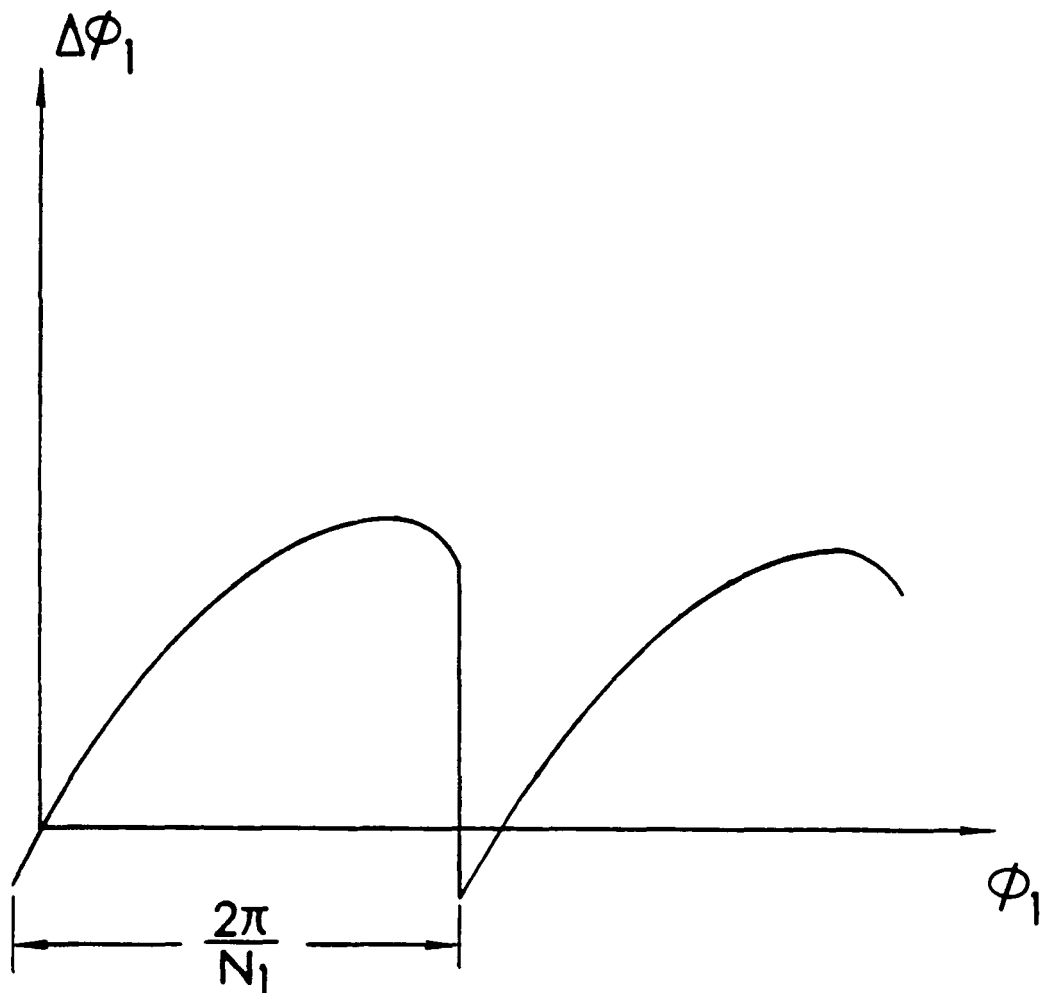


FIG. 2.4 Discontinued Parabolic Function of Transmission Errors

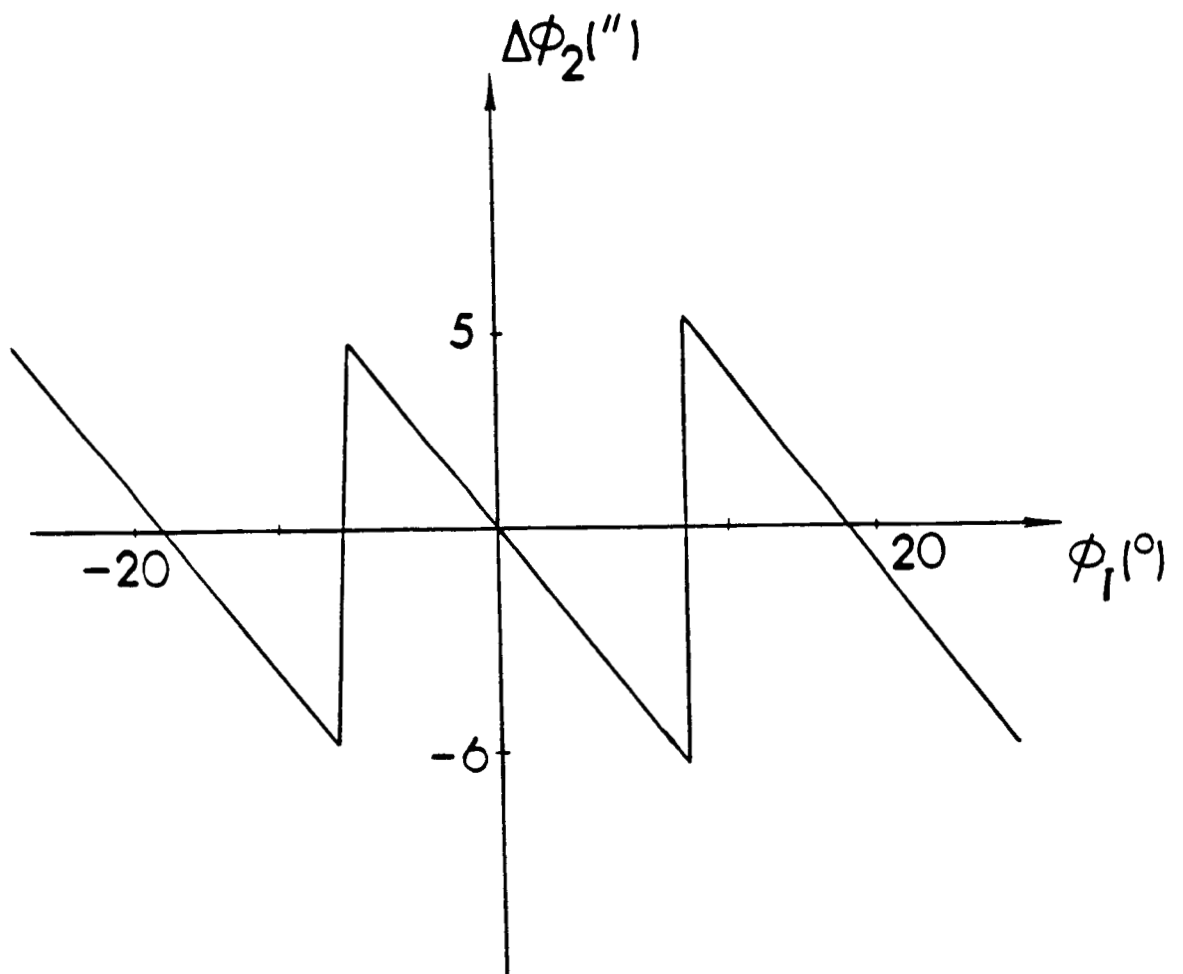


FIG. 2.5 Transmission Error of Helical Gears

$P_n = 10 \text{ in}^{-1}$   
 $N_p = 20$   
 $N_G = 40$   
 $\beta_p = 15^\circ 10'$   
 $\beta_G = 15^\circ$

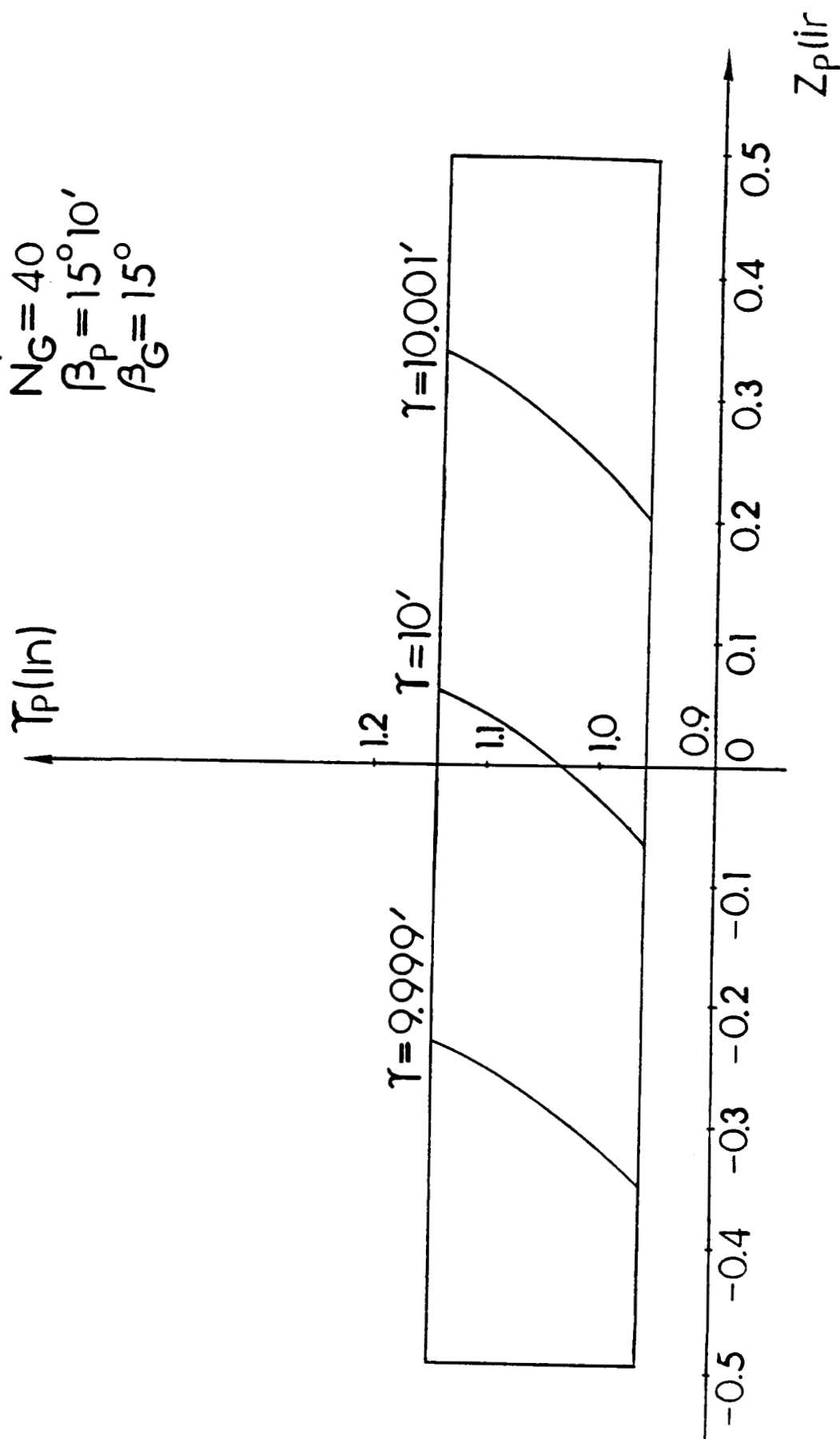


FIG. 2.6 Shift of Contact Path

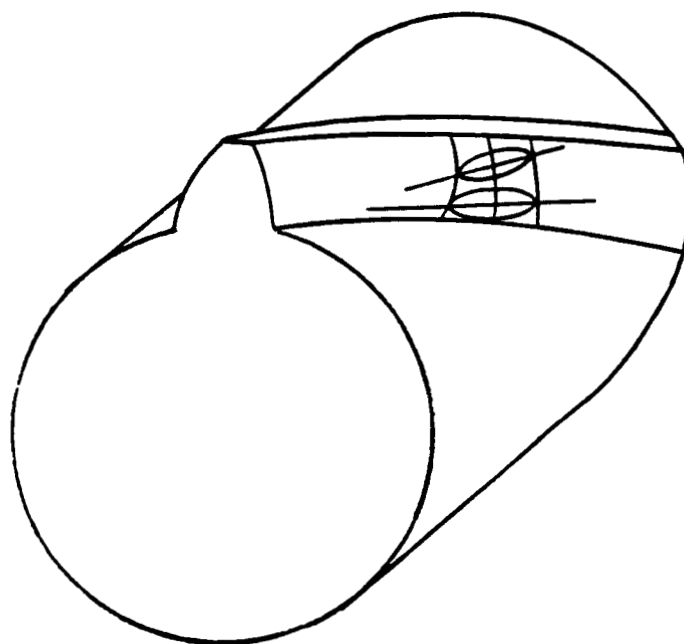
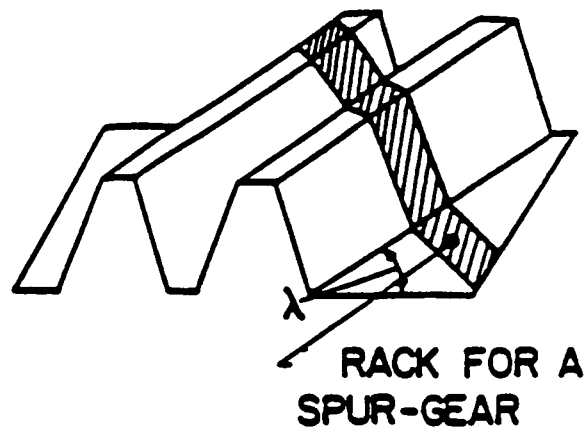
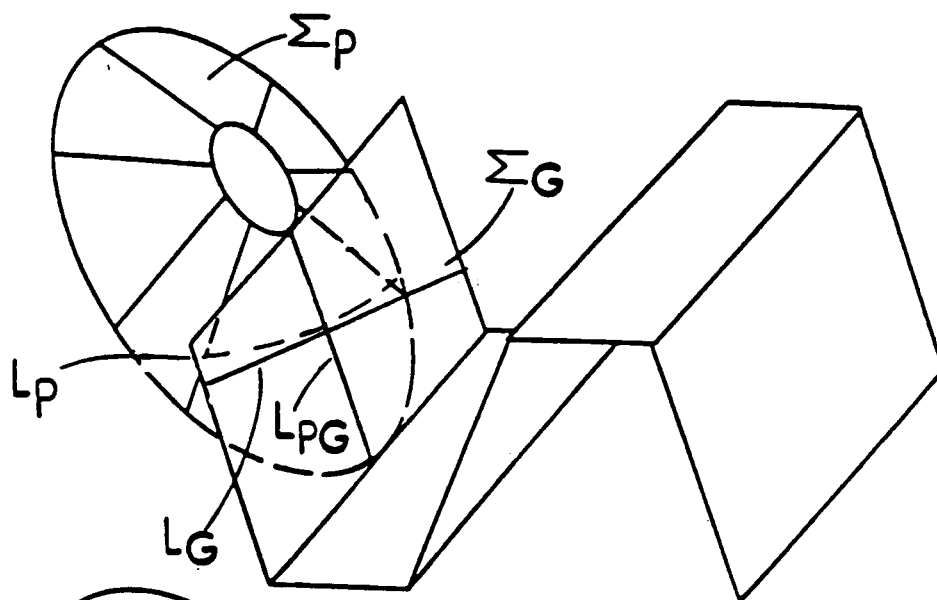


FIG. 3.1 Contact Ellipses on the Pinion Tooth Surface

(a)



(b)



(c)

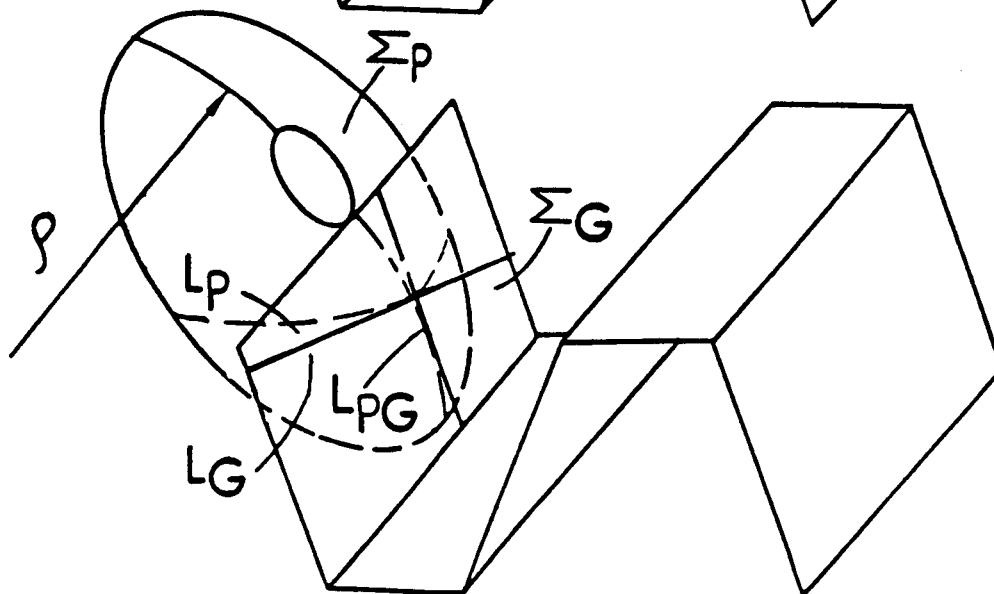


FIG. 3.2 Generating Surfaces

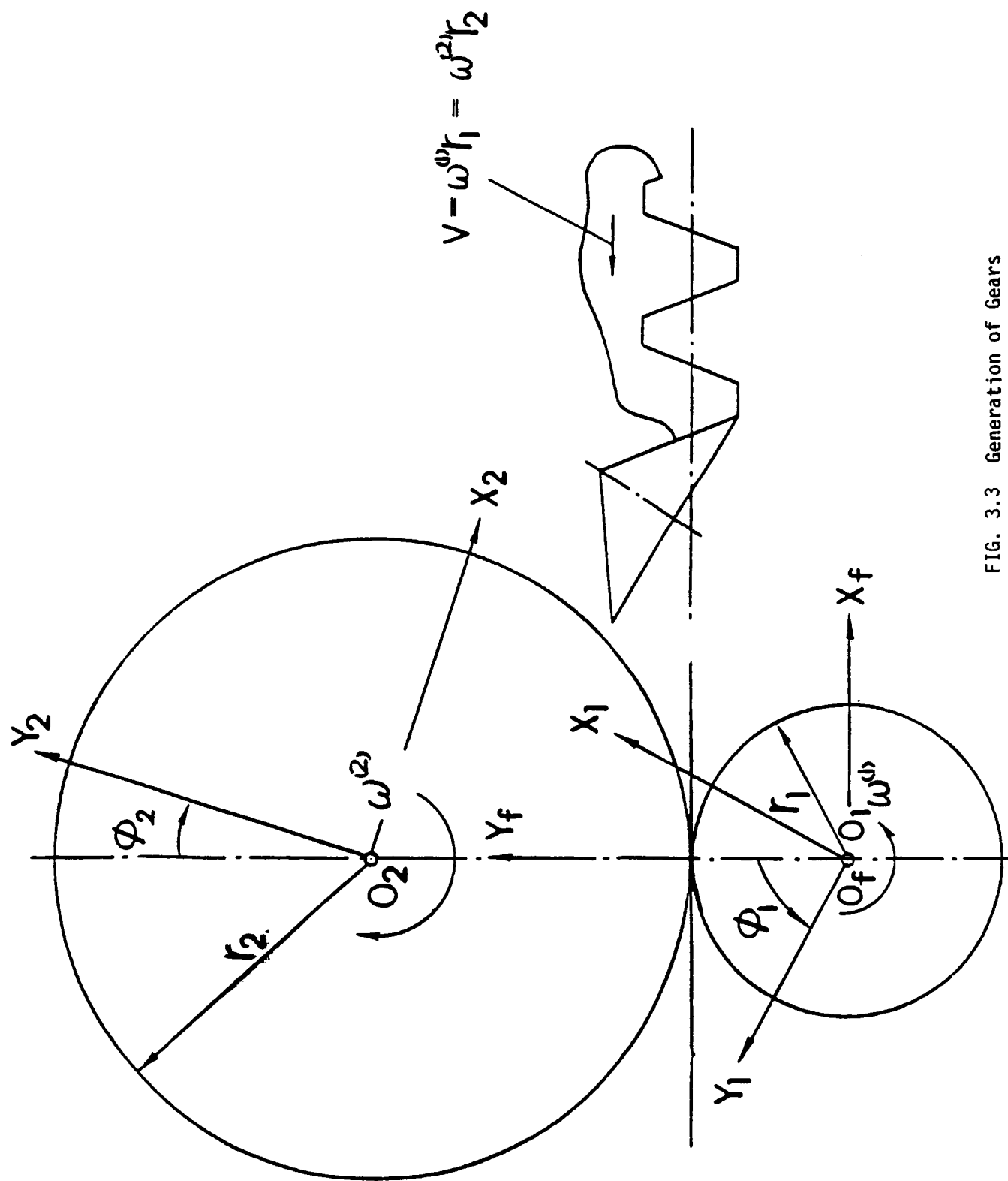


FIG. 3.3 Generation of Gears

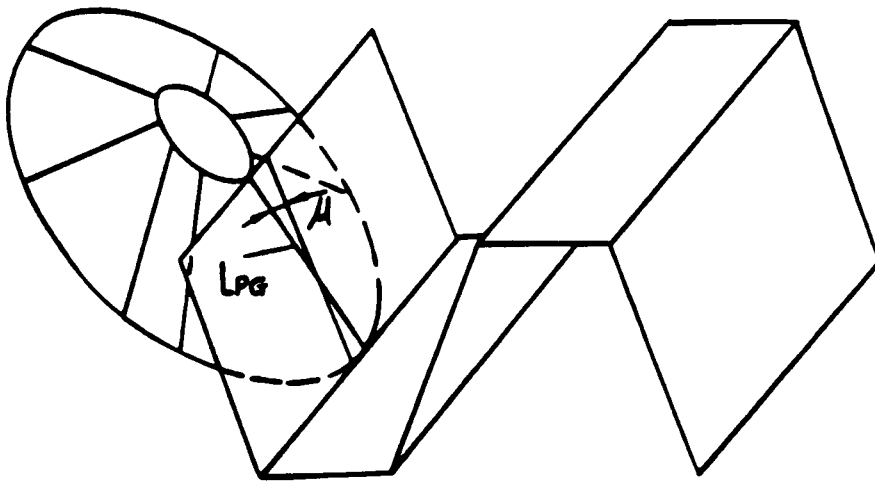


FIG. 3.4 Generating Tool Surfaces

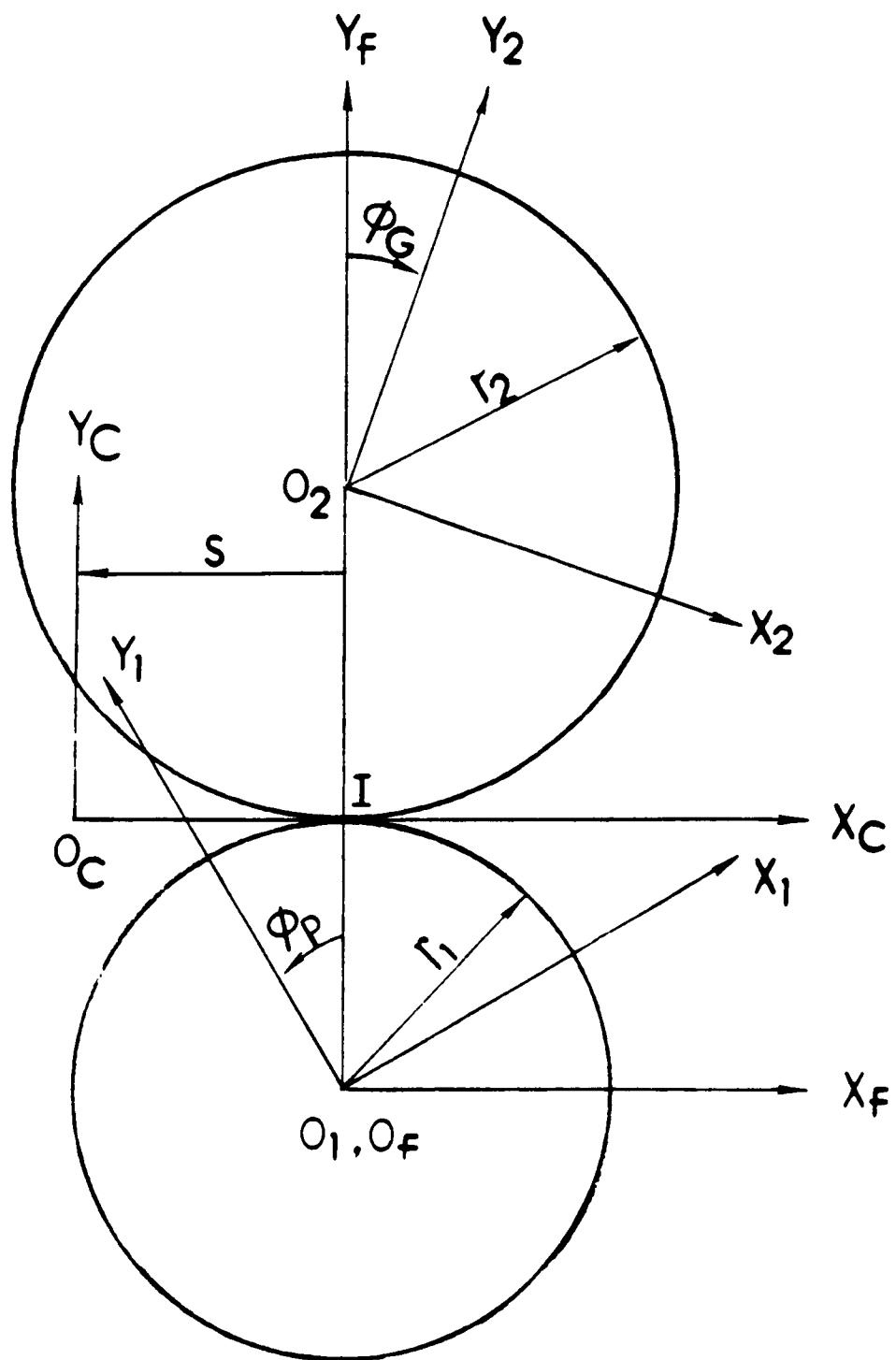
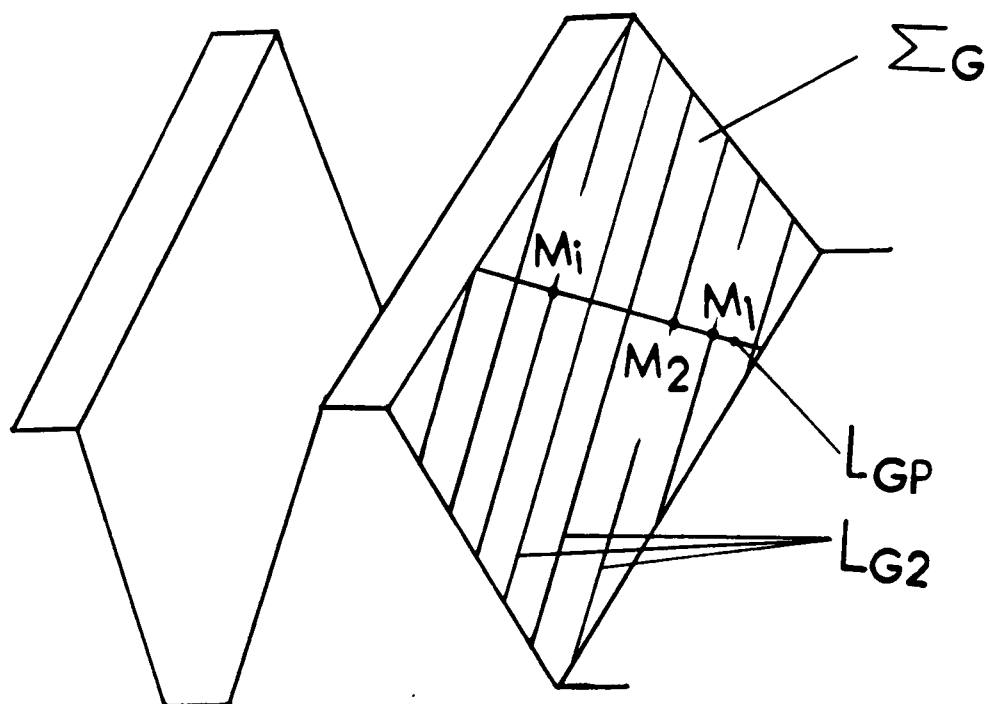


FIG. 3.5 Coordinate Systems for Generating Pinion and Gear Simultaneously

(a)



(b)

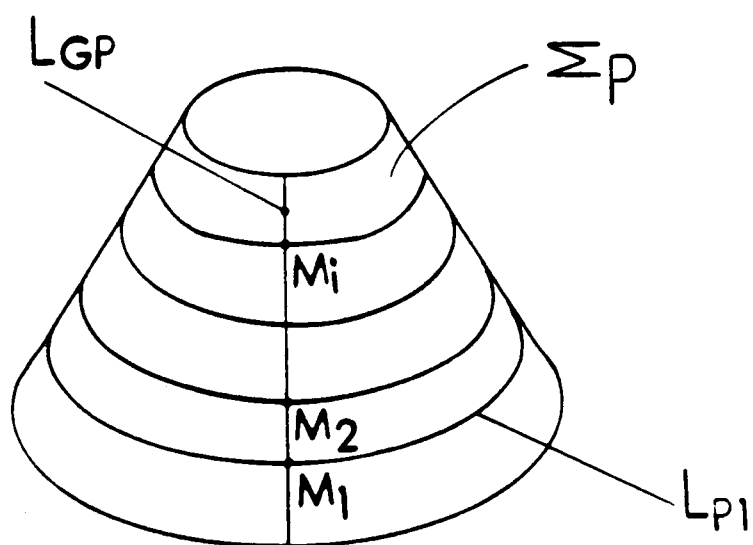


FIG. 3.6 Pinion and Gear Generating Surfaces

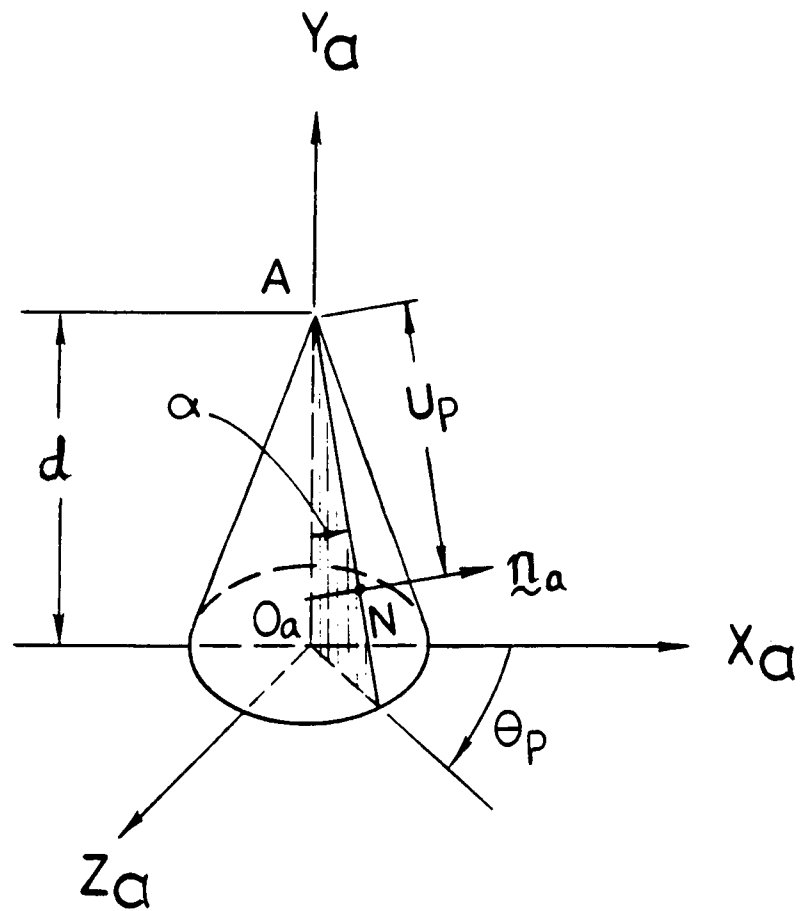


FIG. 3.7 Cone Surface

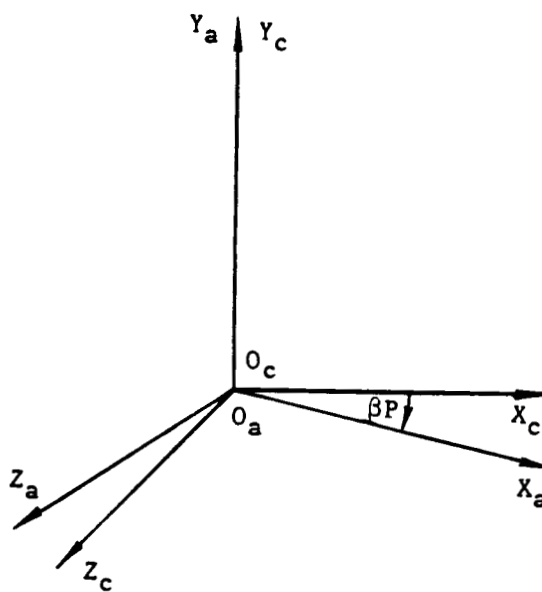
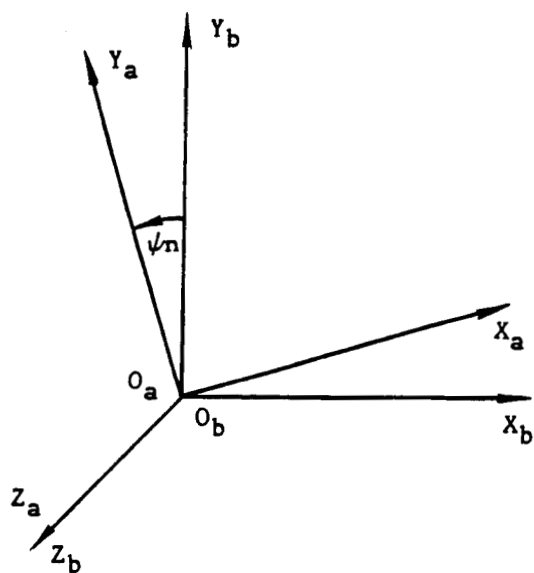
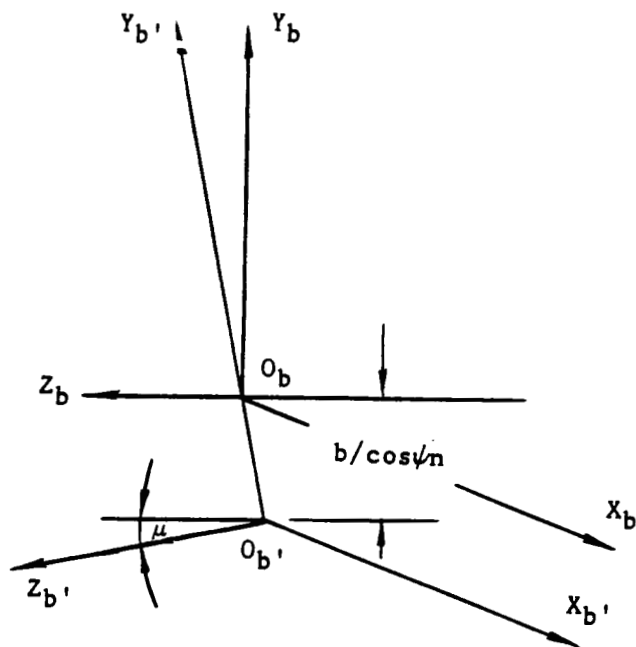
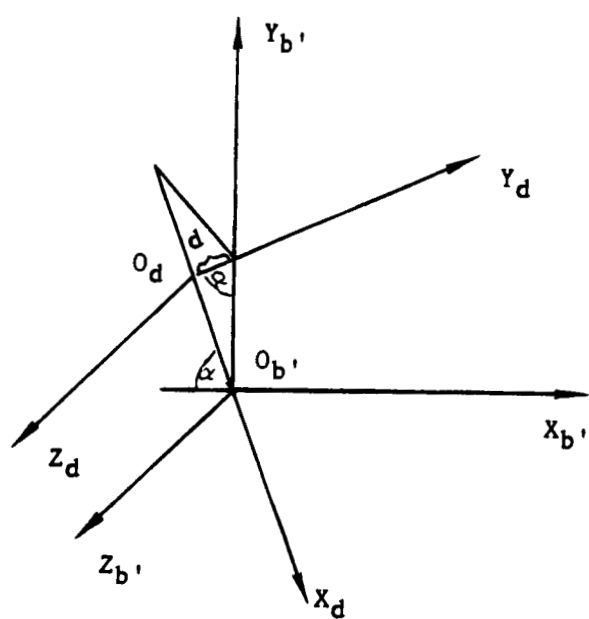


FIG. 3.8 Installment of Conic Tool

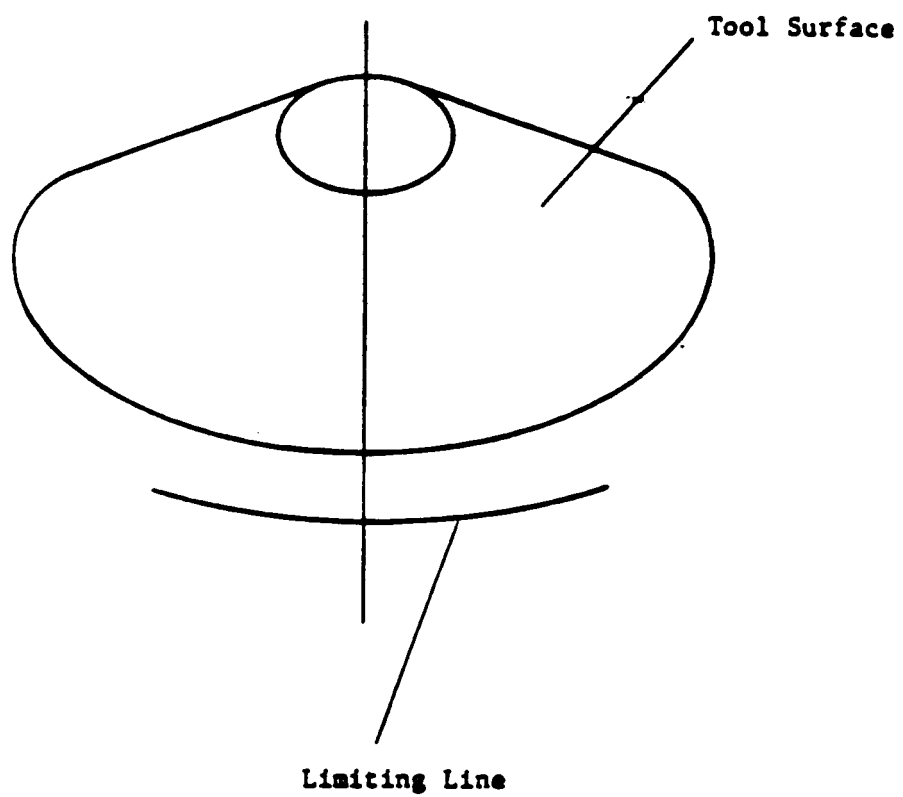


FIG. 3.9 Limiting Line of Tool Surface

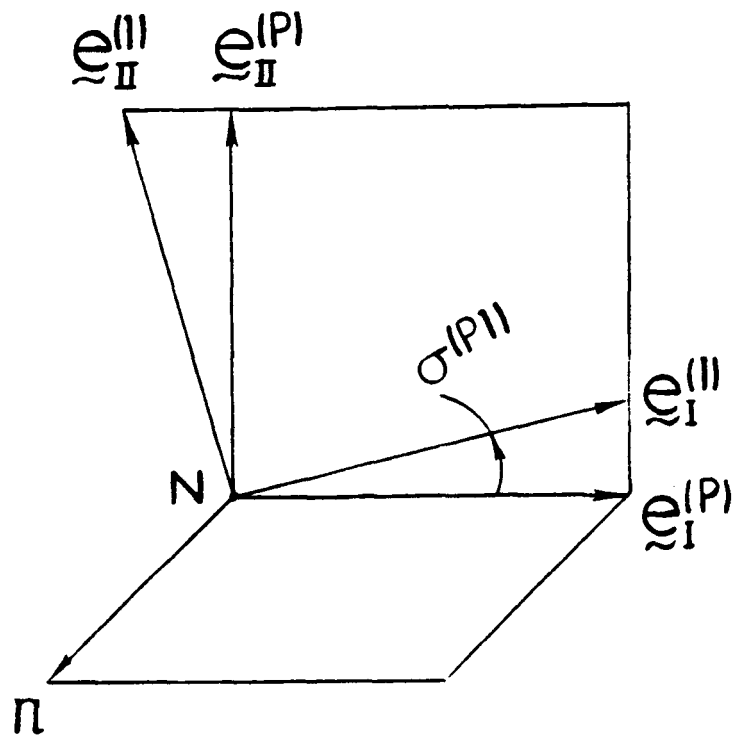


FIG. 3.10 Principal Directions of Generated and Generating Surfaces

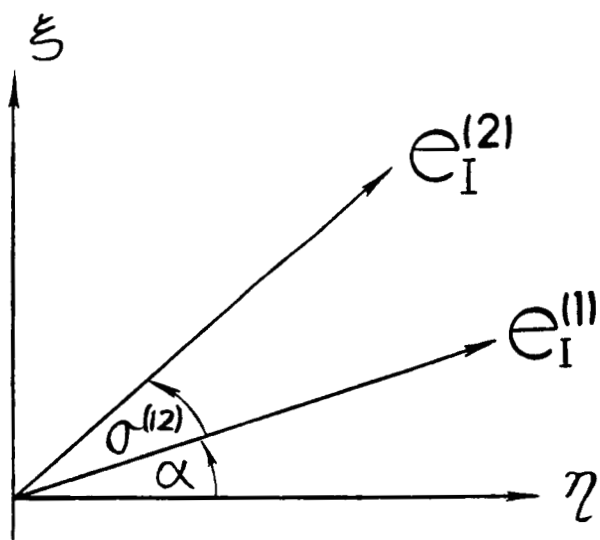
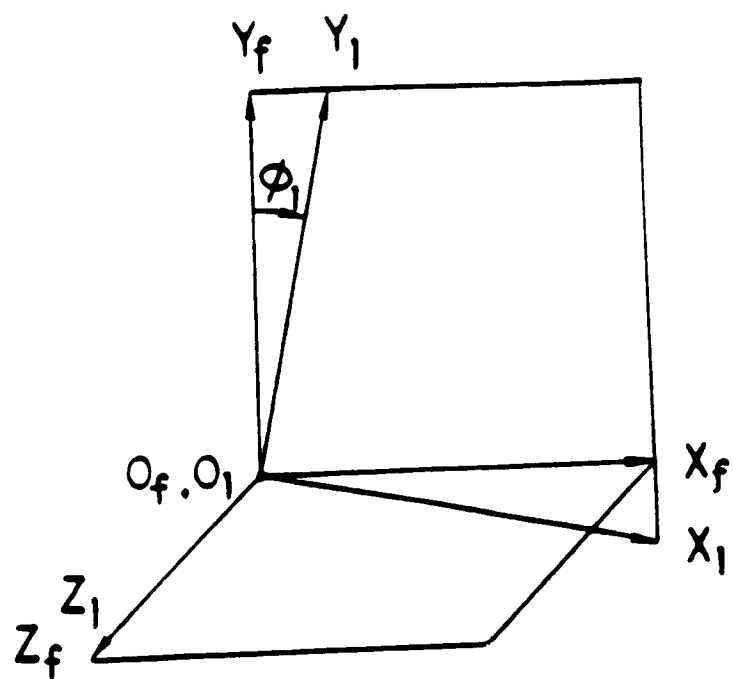


FIG. 3.11 Orientation of Contact Ellipse

(a)



(b)

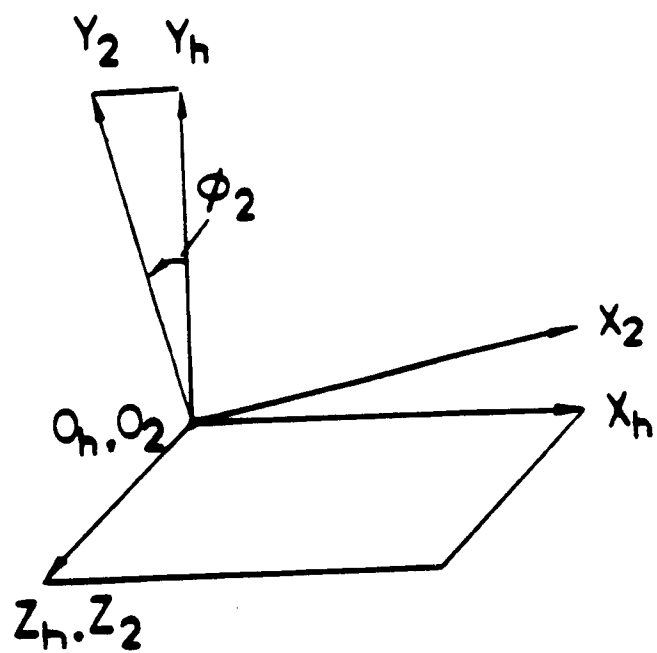


FIG. 3.12 Coordinate Systems for Gear Tooth Rotation

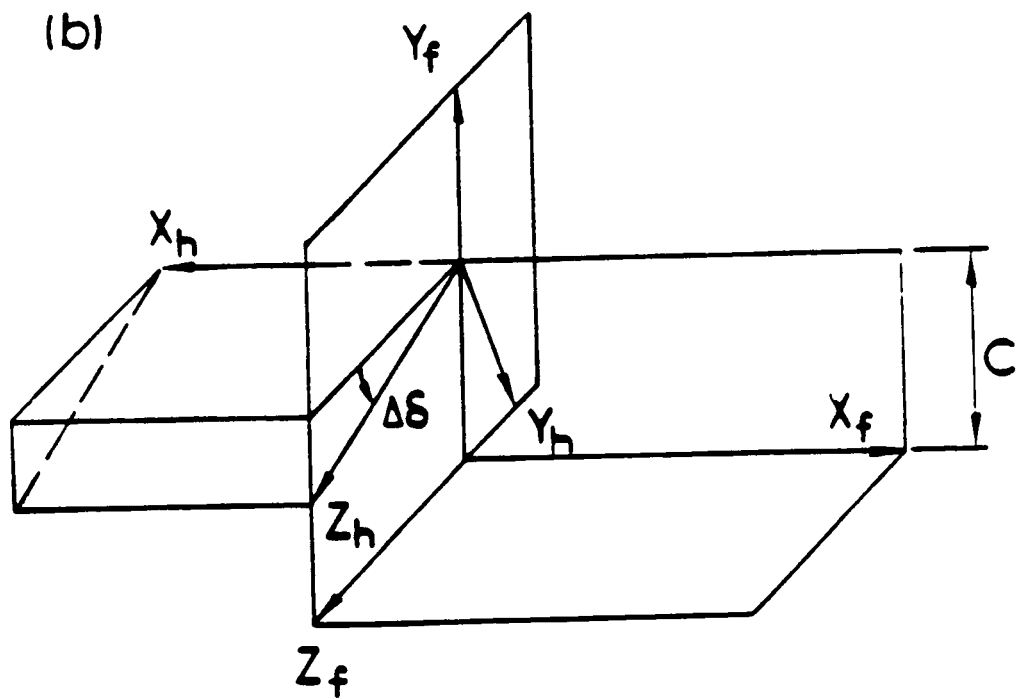
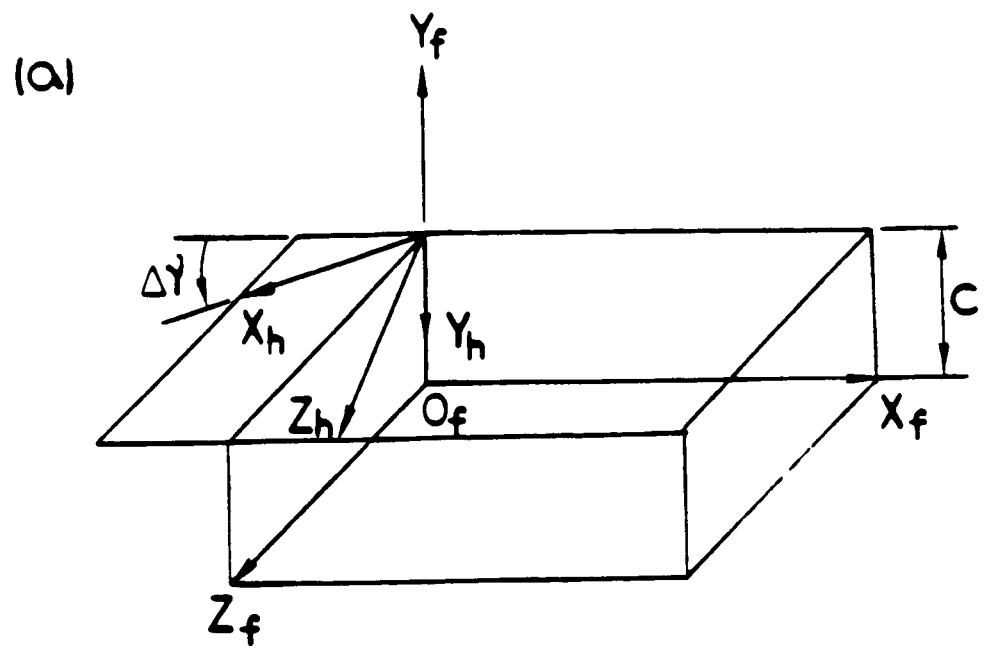


FIG. 3.13 Coordinate Systems for Simulation of Gear Misalignment

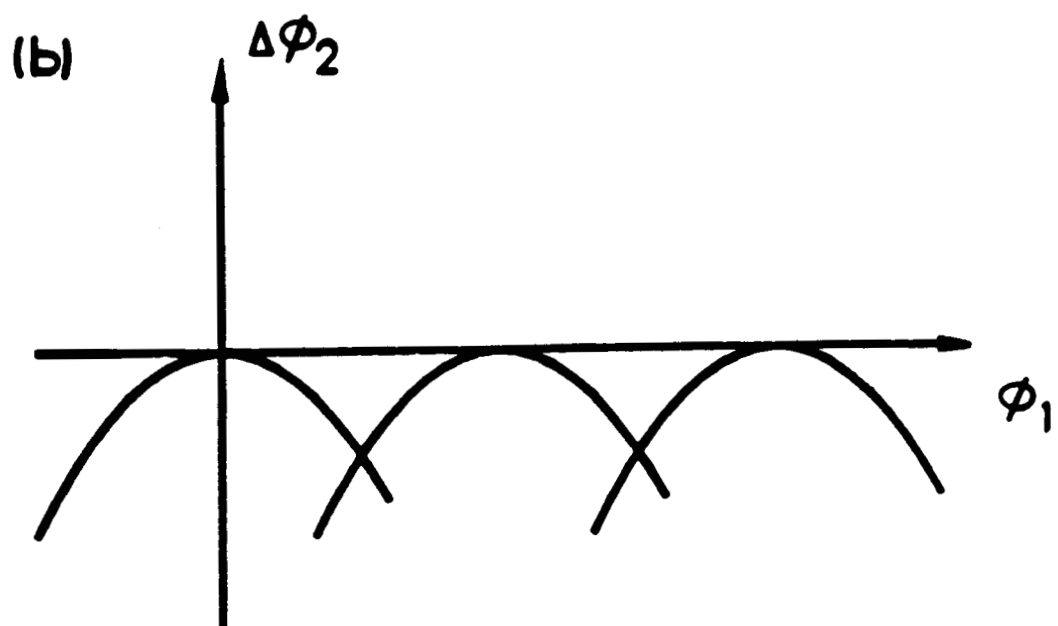
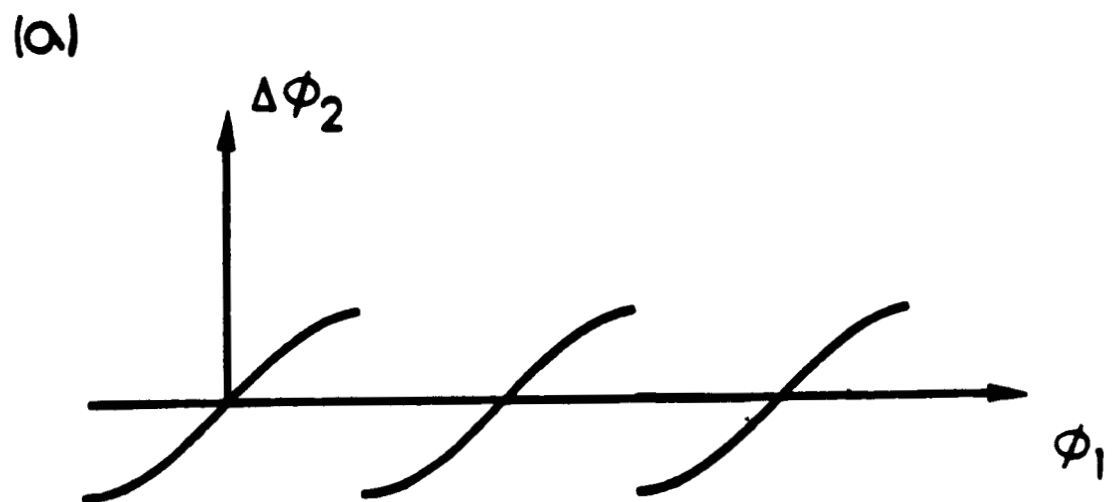


FIG. 3.14 Two Types of Transmission Errors



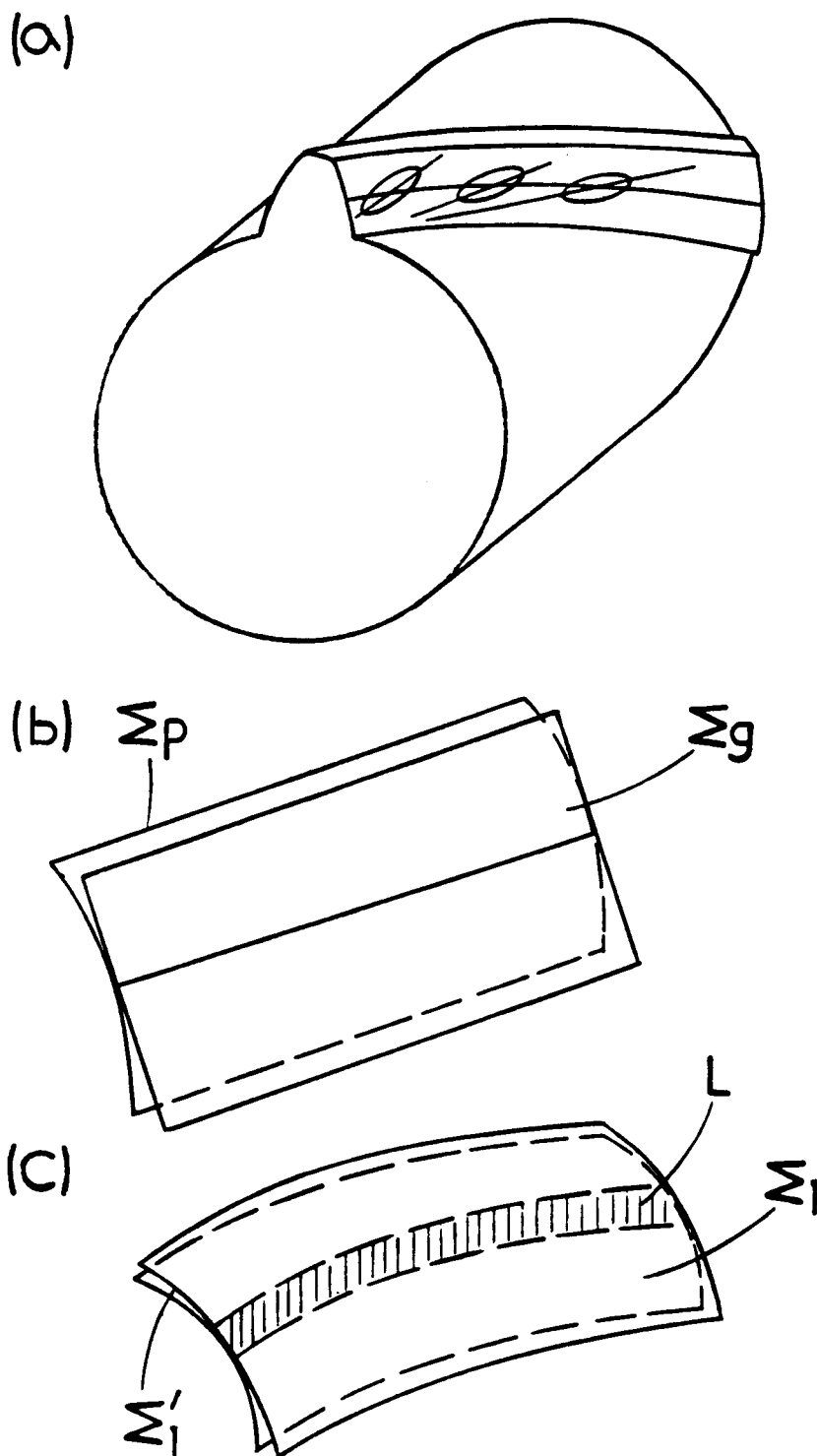


FIG. 4.1 Helical Gear and Generating Tool Surface

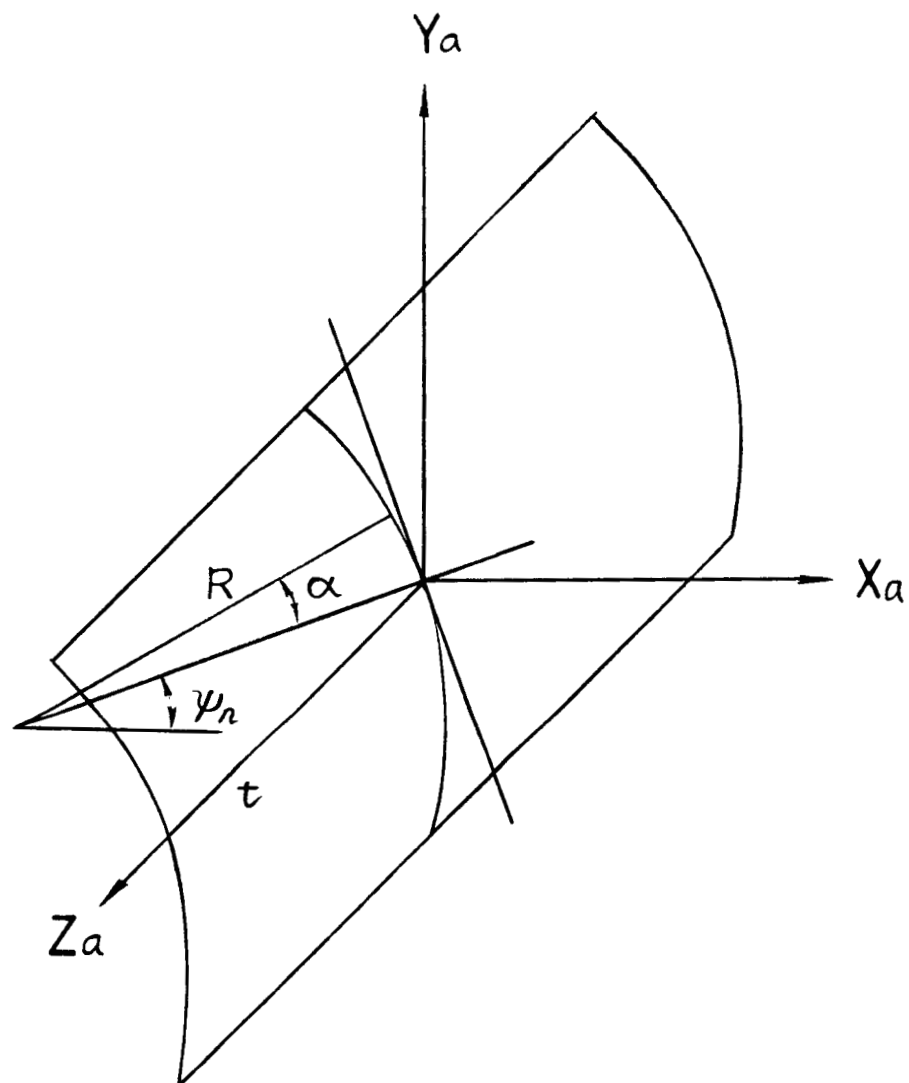


FIG. 4.2 Helical Pinion Generating Surface

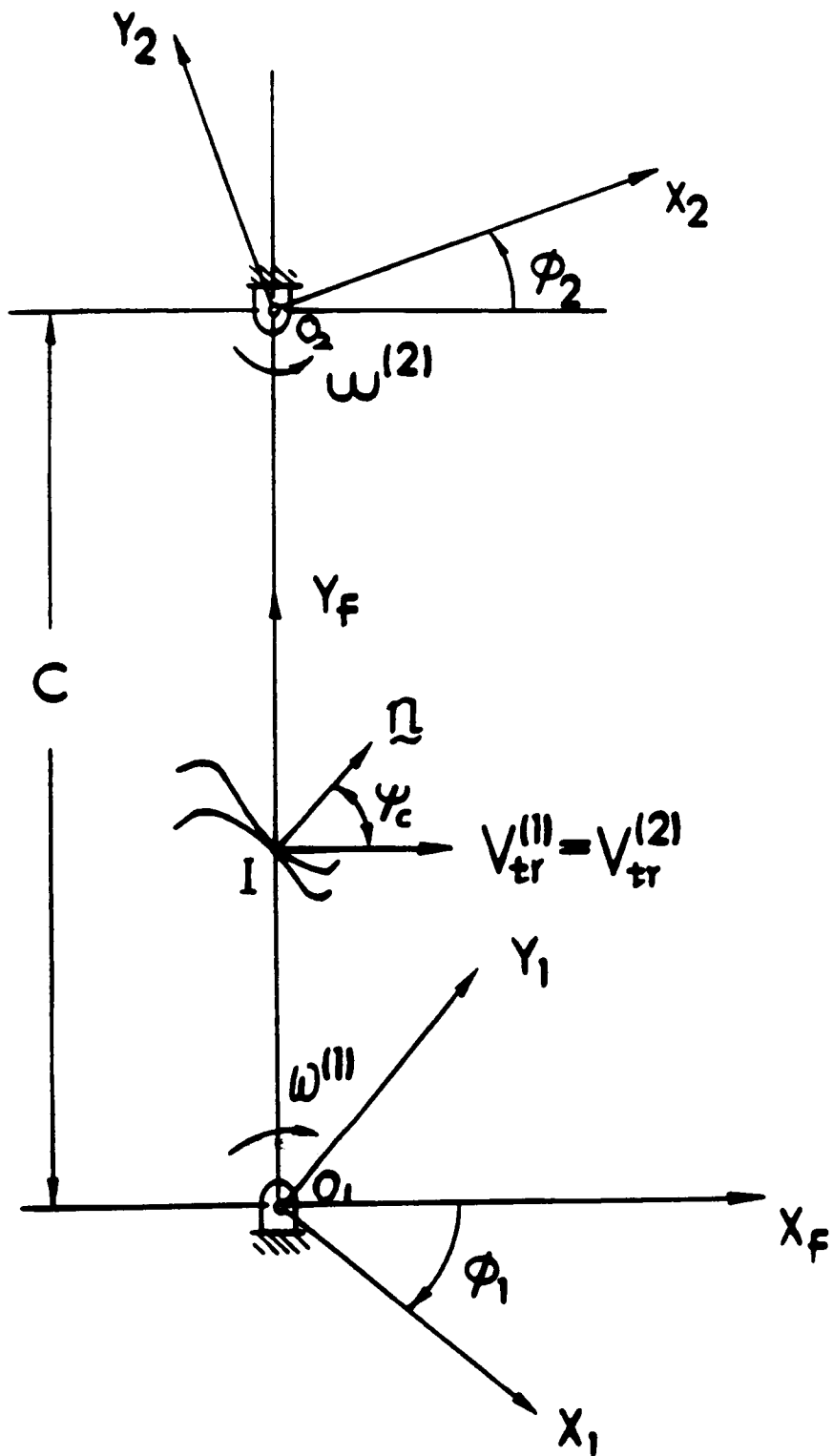


FIG. 4.3 Basic Coordinate Systems for Meshing Gears

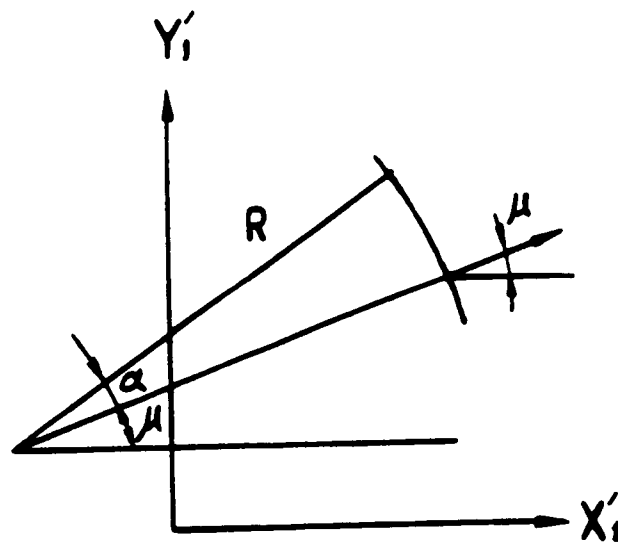


FIG. 4.4 Derivation of Deviated Helical Pinion Tooth Surface

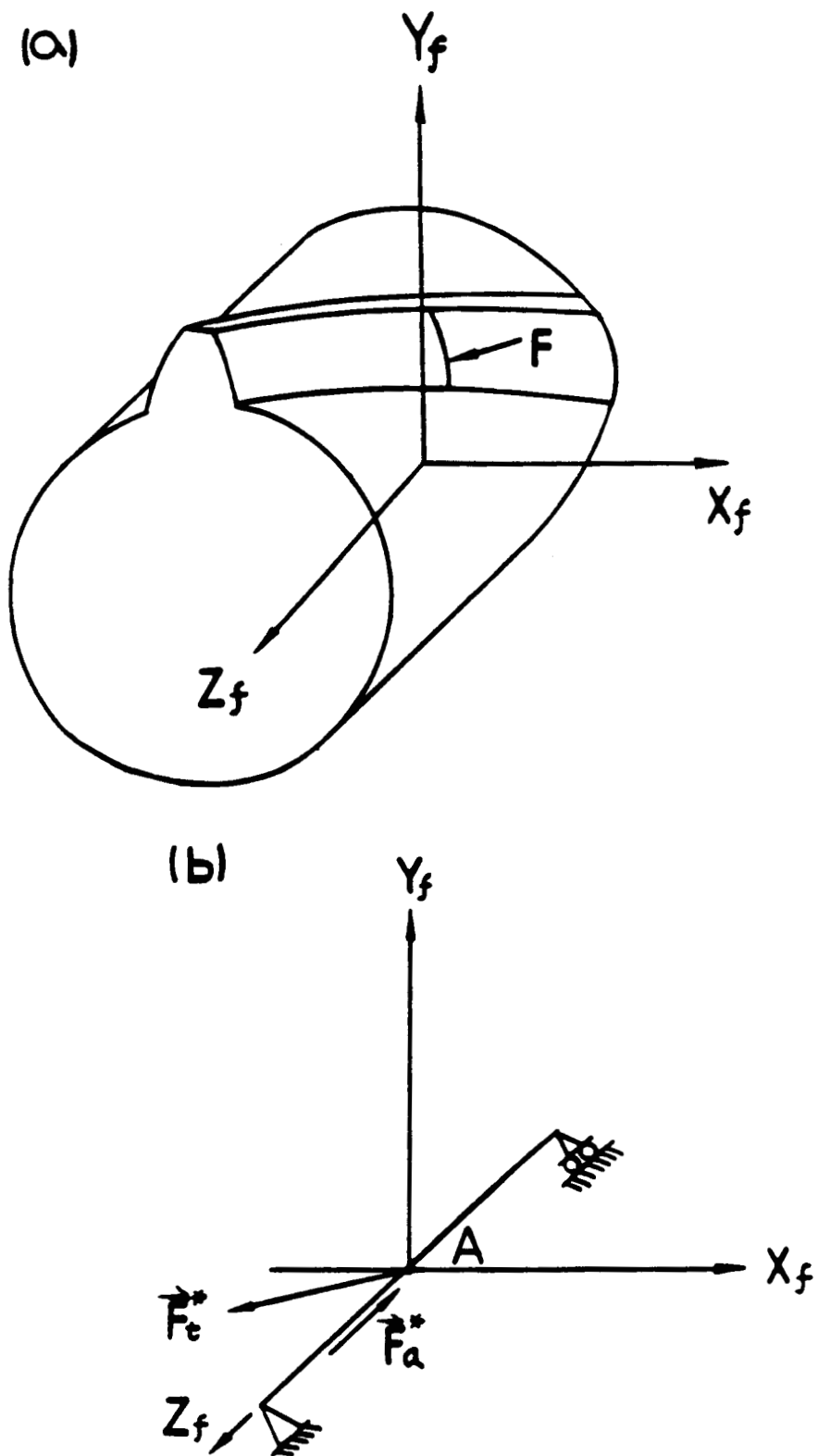


FIG. 5.1 Force Applied on Pinion and Its Shaft

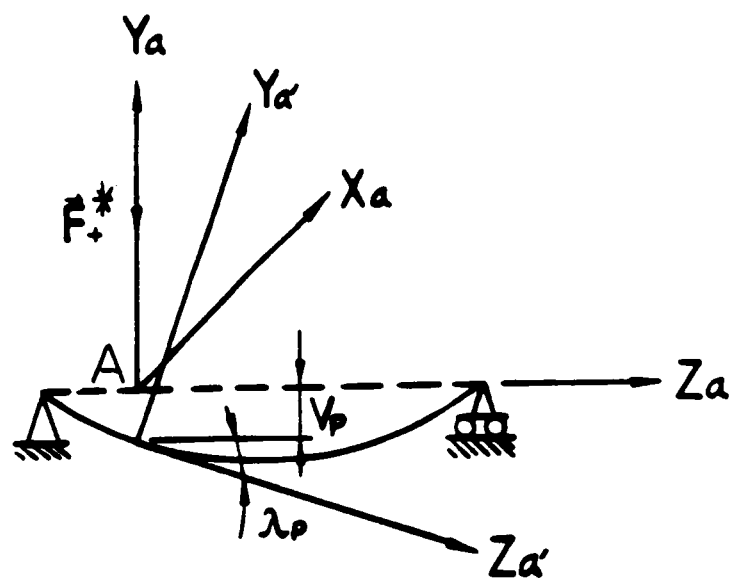


FIG. 5.2 Shaft Deflection

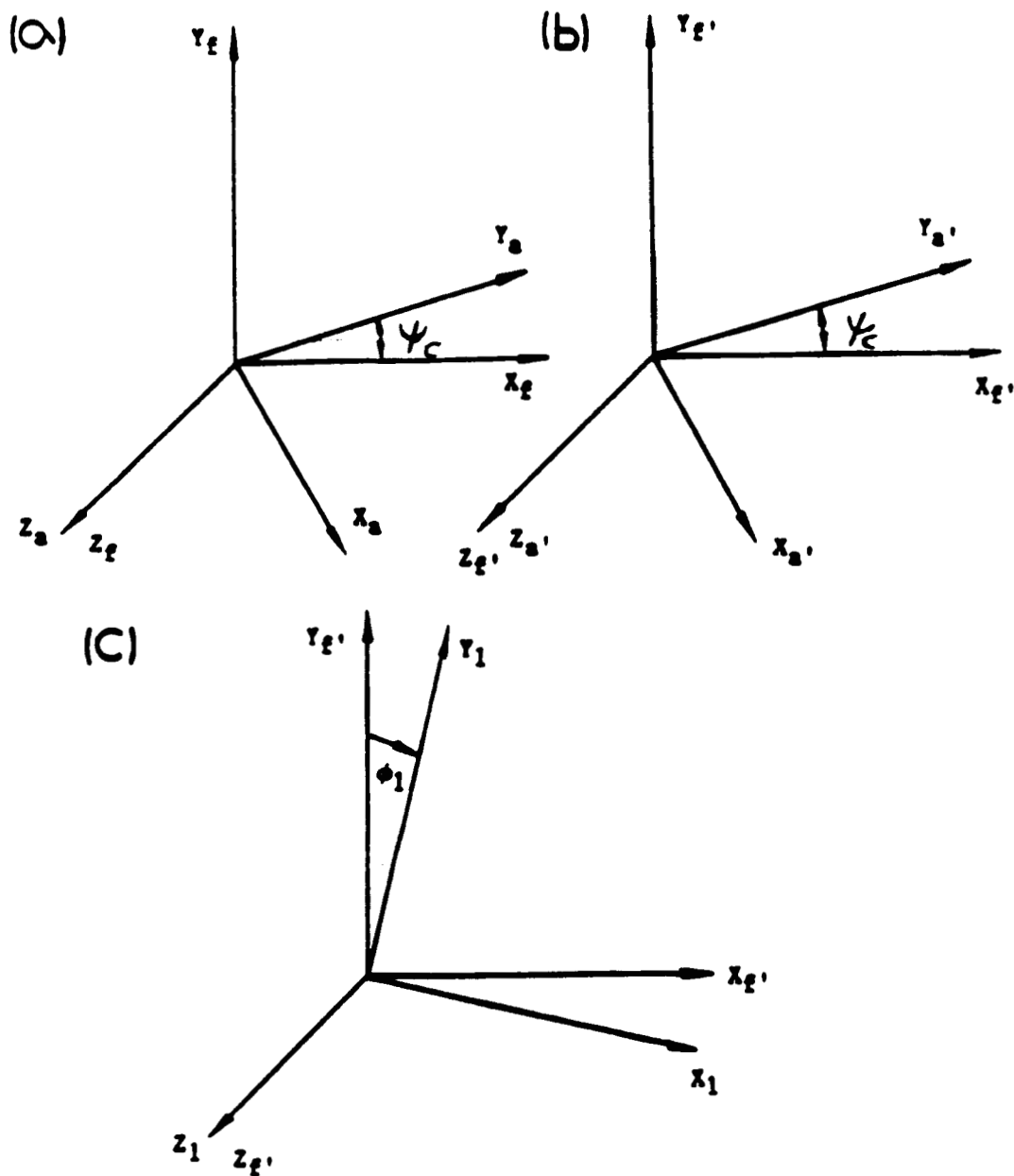


FIG. 5.3 Coordinate Systems for Simulation of Shaft Deformation

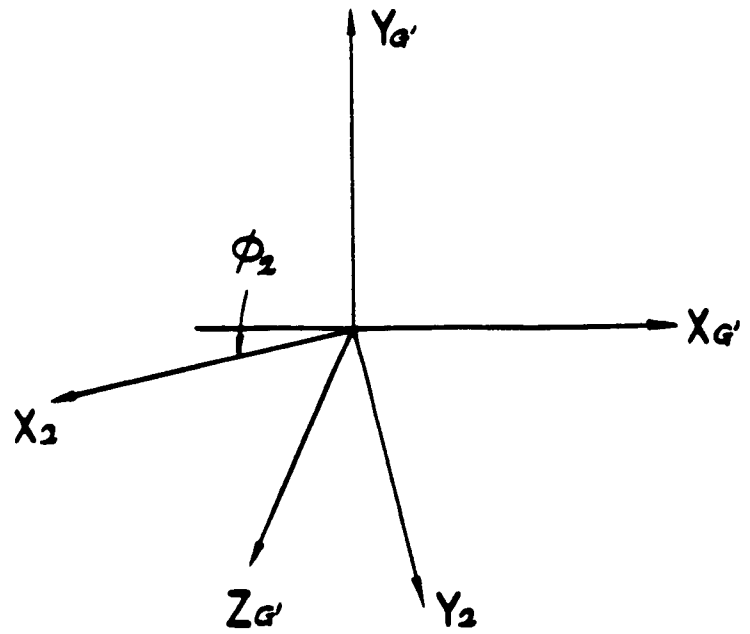


FIG. 5.4 Coordinate Systems for Simulation of Shaft Deformation

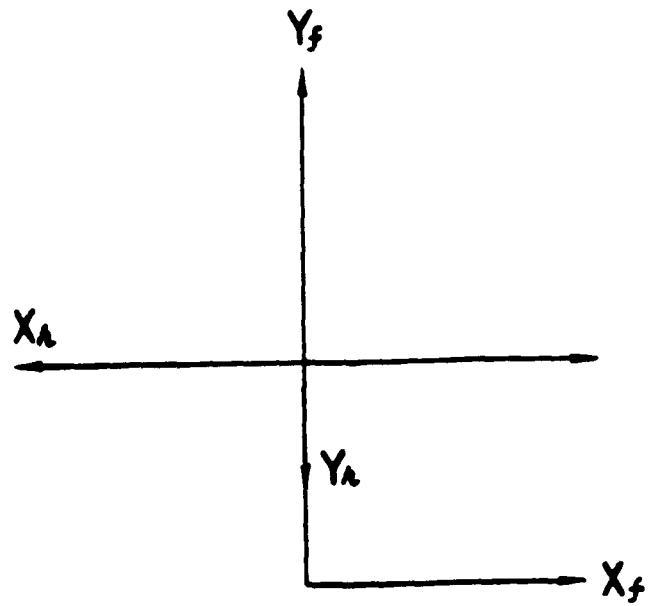
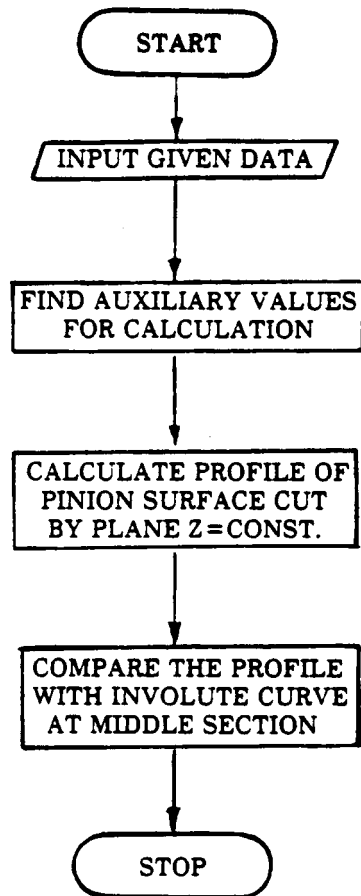


FIG. 5.5 Coordinate Systems for Simulation of Shaft Deformation

## FLOWCHART FOR PROGRAM I



```

C... *****
C... *
C... *
C... *          PROGRAM I
C... * SURFACE OF HELICAL PINION GENERATED BY CONE CUTTER
C... *
C... *          AUTHORS: FAYDOR LITVIN
C... *          JIAO ZHANG
C... *
C... *
C... *****
C
C PURPOSE
C
C THIS PROGRAM IS USED TO CALCULATE THE SURFACE OF A HELICAL PINION
C WHICH IS GENERATED BY CONE CUTTER
C
C NOTE
C
C THIS PROGRAMME IS WRITTEN IN FORTRAN 77. IT CAN BE COMPILED BY V
C COMPILE IN IBM MAINFRAME OR FORTRAN COMPILER IN VAX SYSTEM
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      DOUBLE PRECISION KSIN,MU,KSIC,MUO
C      DIMENSION X(100),Y(100),ERROR(100),FEEREC(100),UREC(100),
C      +          THETAR(100)
C
C DEFINE PARAMETERS USED BY PROGRAMS
C
C (1) IN ANG LP ARE UNIT NUMBERS ASSIGNED TO THE INPUT AND OUTPUT
C DEVICES
C      IN=5
C      LP=6
C (2) NDBUG IS USE TO CONTROL THE AUXILIARY OUTPUT FOR DEBUGGING
C      NDBUG=1
C (3) NSOLVE IS THE UPPER LIMITATION OF REPEATATION FOR SOLVING
C NONLINEAR EQUATIONS;
C EPSI IS THE CLEARANCE OF FUNCTION VALUES WHEN THE FUNCTIONS
C IS CONSIDERED AS SOLVED (ALL FUNCTIONS HAVE FORMS OF F(X)=0);
C DELTA IS THE CLERANCE OF VARIABLE INCREMENT WHEN FUNCTION IS
C SOLVED
C NC, EPSI AND DELTA MAY BE CHANGED WHEN SOLUTIONS ARE DIVERGENT
C OR LESS ACCURATE
C      NSOLVE=100
C      DELTA=1.D-15
C      EPSI=1.D-15
C (4) OTHER PARAMETERS(DON'T CHANGE)
C      DR=DATAN(1.D0)/45.D0
C
C DEFINE INPUT PARAMTERS OF PROBLEM(USE INCH AS UNIT OF LENGTH)
C
C (1) PINION AND GEAR: PN=DIAMETRAL PITCH; N1=PINION TOOTH NUMBER;
C      KSIN=PRESSURE ANGLE IN NORMAL SECTION (DEGREE);
C      BETAP=HELIX ANGLE (DEGREE);
C      HD=HEIGHT OF DEDENDUM OF PINION;

```

```

C      HA=HEIRHT OF ADDENDUM OF PINION
C      ZCOE=COEFFICIENT OF PINION TOOTH LENGTH (THE LENGTH= ZCOE/PN)
      PN=10.D0
      N1=20
      KSIN=20.D0*DR
      BETAP=20.D0*DR
      HD=1.D0/PN
      HA=1.D0/PN
      ZCOE=10.
C      (2) TOOL: ALPHA=HALF OF CONE VERTEX ANGLE(DEGREE);
C      RC=RADIUS OF BOTTOM CIRCLE OF CONE;
C      MU=TILT ANGLE TO INSTALL PINION-CUTTING TOOL
      ALP=80.D0*DR
      MU0=DATAN(DSIN(KSIN)*DTAN(BETAP))
      MU=1.*MU0
      RC=1.D0
C      (3) OUTPUT:
C      NZ=NO. OF CROSS SECTIONS WHERE PINION PROFILE IS SIMULATED;
C      NU=NO. OF MAXIMUM POINTS USED FOR SIMULATE PINION PROFILE
C      UIN=INCREMENT OF PINION TOOTH SURFACE COORDINATE U
      NZ=21
      NU=61
      UIN=2.5D0*CH/(NU-1)
C
C      DESCRIPTION OF OUTPUT VARIABLES
C      Z1=DISTANCE BETWEEN CROSS SECTION CONSIDERED AND MIDDLE CROSS
C      SECTION
C      NO=OUTPUT NO.
C      U=TOOL SURFACE COORDINATE
C      THETA=TOOL SURFACE COORDINATE
C      FEE=PINION TOOTH SURFACE GENERATION PARAMETER
C      X1=X COORDINATE OF PINION PROFILE
C      Y1=Y COORDINATE OF PINION PROFILE
C      R1=RADIUS OF PINION PROFILE
C      VSH=AVERAGE DEVIATION SHIFT OF CROSS SECTION PROFILE FROM PROFILE
C      OF GENERAL PINION (INVOLUTE CURVE)
C      VPE=MAXIMUM DEVIATION OF CROSS SECTION PROFILE FROM INVOLUTE CURVE
C      VSD=STANDARD DEVIATION OF CROSS SECTION PROFILE FROM INVOLUTE
C      CURVE
C
C      THE PROGRAM IS WRITTEN BY JIAO ZHANG
      SK=DSIN(KSIN)
      CK=DCOS(KSIN)
      SB=DSIN(BETAP)
      CB=DCOS(BETAP)
      SM=DSIN(MU)
      CM=DCOS(MU)
      SA=DSIN(ALPHA)
      CA=DCOS(ALPHA)
      KSIC=DATAN(SK/CK/CB)
      CKC=DCOS(KSIC)
      SKC=DSIN(KSIC)
      PT=PN*CB
      RP=N1/2./PT

```

```

RB=RP*DCOS(KSIC)
RA=RP+HA
CL=RC/SA
CH=HD/CK/CM
A1=CA*CK*CB+SA*SK*CM*CB+SA*SM*SB
A2=SK*SM*CB-CM*SB
A3=CA*SK*CM*CB-SA*CK*CB+CA*SM*SB
A4=SK*CM*CB+SM*SB
B1=CA*SK-SA*CK*CM
B2=CK*SM
B3=SA*SK+CA*CK*CM
B4=CK*CM
C1=CA*CK*SB+SA*SK*CM*SB-SA*SM*CB
C2=SK*SM*SB+CM*CB
C3=CA*SK*CM*SB-SA*CK*SB-CA*SM*CB
C4=-SK*CM*SB+SM*CB
D1=CM*CB+SK*SM*SB
D2=SA*SM*CB-SB*(CA*CK+SA*SK*CM)
D3=CA*CK*SB
D4=CA*(CA*SK-SA*CK*CM)
D5=SA*(SA*SK+CA*CK*CM)
D6=CA*CK*SM
DO 5 I=1,NZ
Z1=-ZCOE/PN/2.+ZCOE/PN*FLOAT(I-1)/FLOAT(NZ-1)
FEE0=Z1*SB/CB/RP
CF0=DCOS(FEE0)
SF0=DSIN(FEE0)
R1=RP
ERR=0.
NP=0
DO 15 J=1,NU
IF (R1.GT.RA.AND.J.GT.1) GOTO 55
U=CL-UI*FLOAT(J-1)
TEMP1=(Z1-(CL-CH)*C4-U*CA*C3)/U/SA/DSQRT(C1*C1+C2*C2)
TEMP2=DARSIN(C1/DSQRT(C1*C1+C2*C2))
TEMP3=DARSIN(TEMP1)
THETA=TEMP3-TEMP2
CT=DCOS(THETA)
ST=DSIN(THETA)
XC=U*SA*(CT*A1+ST*A2)+U*CA*A3-(CL-CH)*A4
YC=U*SA*(CT*B1-ST*B2)-U*CA*B3+(CL-CH)*B4
FEE=(U*(CT*D1+ST*D2)-(CL-CH)*((CT*CA*CA+SA*SA)*D1-ST*D3))/RP/
+ (CT*D4+D5-ST*D6)
CF=DCOS(FEE)
SF=DSIN(FEE)
X1=XC*CF+YC*SF-RP*FEE*CF+RP*SF
Y1=-XC*SF+YC*CF+RP*FEE*SF+RP*CF
R1=DSQRT(X1*X1+Y1*Y1)
IF (R1.LT.RB) GOTO 15
NP=NP+1
X(NP)=X1
Y(NP)=Y1
FEEREC(NP)=FEE
UREC(NP)=U

```

```

    THETAR (NP)=THETA
    XE=X1*CFO+Y1*SFO
    YE=-X1*SFO+Y1*CFO
    YS=YE
    FEET=FEE+FEE0
    DO 25 K=1, NSOLVE
    W=RP*FEET*CKC*DSIN (FEET-KSIC)+RP*DCOS (FEET)-YS
    DWDF=-RP*SKC*DCOS (FEET-KSIC)+RP*FEET*CK*DCOS (FEET-KSIC)
    DF=-W/DWDF
    IF (DABS (W) .LT. EPSI .AND. DABS (DF) .LT. DELTA) GOTO 65
    FEET=FEET+DF
25  CONTINUE
65  CONTINUE
    XS=-RP*FEET*CKC*DCOS (FEET-KSIC)+RP*DSIN (FEET)
    ERROR (NP)=XE-XS
C    WRITE (LP, 110) K, W, DF, ERROR (NP), FEET, FEE+FEE0
C 110 FORMAT (1X, 'K=', I5, 5X, 'W=', D15.7, 5X, 'DF=', D15.7, D15.7, 2D25.17)
    ERR=ERR+ERROR (NP)
15  CONTINUE
55  VSH=-ERR/FLOAT (NP)
    VPE=0.
    VSD=0.
    DO 35 J=1, NP
    ETEMP=ERROR (J)+VSH
    IF (DABS (ETEMP) .GT. VPE) VPE=DABS (ETEMP)
    VSD=VSD+ETEMP**2
35  CONTINUE
    VSD=DSQRT (VSD/FLOAT (NP))
    WRITE (LP, 10) Z1, VSH, VPE, VSD
10  FORMAT (1H1, /// 'Z1=', F15.7, 5X, 'VSH=', F15.7, 5X, 'VPE=', F15.7, 5X,
+          'VSD', D15.7, 5X/2X, 'NO.', 7X, 'U', 14X, 'THETA', 10X, 'FEE', 12X,
+          'X1', 13X, 'Y1', 13X, 'R1', 13X, 'ERROR')
    DO 45 J=1, NP
    R1=DSQRT (X (J)**2+Y (J)**2)
    ETEMP=ERROR (J)+VSH
    WRITE (LP, 20) J, UREC (J), THETAR (J), FEEREC (J), X (J), Y (J), R1, ETEMP
20  FORMAT (2X, I2, 2X, 7F15.7)
45  CONTINUE
5  CONTINUE
    STOP
    END
$ENTRY
C  FIND AUXILIARY VALUES FOR CALCULATION
    RP=FLOAT (N1)/2.D0/PN
    CL=RC/DSIN (ALP)
    D=CL*DCOS (ALP)
    A1=CL-HDC/DCOS (RKS)
    RPU=RP+HAC
    DO 5 I=1, NL
    Z=FLOAT (I-1)*ZI
    YY=D-Z/DTAN (ALP)
    WRITE (LP, 10) Z
10  FORMAT (1H1/1X, 'Z1=', F15.7/1X, 'NO', 10X, 'Y1', 13X, 'XP', 13X, 'YP',
+          13X, 'R1', 13X, 'FEE')

```

C CALCULATE THE PROPILE OF THE SURFACE CUT BY PLANE Z=CONST

```

KKK=0
DO 15 J=1,N
Y1=YY*FLOAT(J-1)/FLOAT(N-1)
F=DSQRT(1.-(Z/(D-Y1)/DTAN(ALP))**2)
FEE=((D-Y1)*F/DCOS(ALP)-A1*(F*DCOS(ALP)*DCOS(ALP)+DSIN(ALP)*DSIN(
+   ALP)))/RP/(DSIN(ALP)*DCOS(ALP-RKS)-F*DCOS(ALP)*DSIN(ALP-RKS))
XP(I,J)=(D-Y1)*(DTAN(ALP)*F*DCOS(ALP-RKS+FEE)-DSIN(ALP-RKS+FEE))
+   +A1*DSIN(FEE-RKS)-RP*FEE*DCOS(FEE)+RP*DSIN(FEE)
YP(I,J)=-(D-Y1)*(DTAN(ALP)*F*DSIN(ALP-RKS+FEE)+DCOS(ALP-RKS+FEE))
+   +A1*DCOS(FEE-RKS)+RP*FEE*DSIN(FEE)+RP*DCOS(FEE)
R1=DSQRT(XP(I,J)**2+YP(I,J)**2)
IF (R1.GT.RPU) THEN
JJ=J-1
XP(I,J)=XP(I,JJ)+(XP(I,J)-XP(I,JJ))/(R1-R1TEMP)*(RPU-R1TEMP)
YP(I,J)=YP(I,JJ)+(YP(I,J)-YP(I,JJ))/(R1-R1TEMP)*(RPU-R1TEMP)
FEE=FEETEM+(FEE-FEETEM)/(R1-R1TEMP)*(RPU-R1TEMP)
Y1=Y1TEM+(Y1-Y1TEM)/(R1-R1TEMP)*(RPU-R1TEMP)
R1=DSQRT(XP(I,J)**2+YP(I,J)**2)
KKK=1
END IF
WRITE (LP,20) J,Y1,XP(I,J),YP(I,J),R1,FEE
20 FORMAT (1X,I4,4F15.7,F15.7)
IF (KKK.GT.0) GO TO 30
FEETEM=FEE
Y1TEM=Y1
R1TEMP=R1
15 CONTINUE
30 NS(I)=J-1
IF (I.NE.1) GO TO 55
NS1=NS(1)
GO TO 5

```

C PREPARATION OF INTERPLORATION

```

55 NS2=NS(I)
DO 105 L=1,NS2
J=NS2+1-L
IF (YP(1,NS1).GT.YP(I,J)) GO TO 110
105 CONTINUE
110 NS2=J
NREC=2
XERS=0.D0
DO 115 L=1,NS2
DO 125 J=NREC,NS1
IF (YP(1,J).GT.YP(I,L)) GO TO 120
125 CONTINUE
120 NREC=J
J1=J-1
XERROR(L)=XP(I,L)-XP(1,J1)-(YP(I,L)-YP(1,J1))*(XP(1,J)-XP(1,J1))
+   / (YP(1,J)-YP(1,J1))
XERS=XERS+XERROR(L)
115 CONTINUE
XERS=-XERS/FLOAT(NS2)
XPE=0.D0
SDX=0.D0

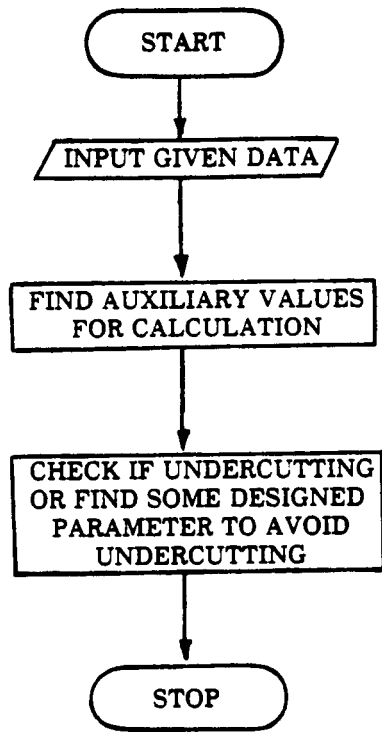
```

```

      IF (NDEBUG.GT.2) WRITE (LP,80)
80  FORMAT (1H1,13X,'NO.',4X,'DIVIATION VALUE')
      DO 45 L=1,NS2
      XERROR(L)=XERROR(L)+XERS
      IF (NDEBUG.GT.2) WRITE (LP,50) L,XERROR(L)
50  FORMAT (13X,I3,2F15.7)
      IF (DABS(XERROR(L)).GT.XPE) XPE=DABS(XERROR(L))
      SDX=SDX+XERROR(L)**2
45  CONTINUE
      SDX=DSQRT(SDX/FLOAT(NS2))
      WRITE (LP,40) XERS,XPE,SDX
40  FORMAT (///1X,'XSH=',E15.7,5X,'XPE=',E15.7,5X,'SDX=',E15.7)
      IF (NDEBUG.GT.2) WRITE (LP,60) NS1,NS2
60  FORMAT (//5X,'NS1=',I6,6X,'NS2=',I6)
      5  CONTINUE
      STOP
      END

```

## FLOWCHART FOR PROGRAM II



```

C... *****
C... *
C... *
C... *          PROGRAM II
C... *          UDDERCUTTING CONDITION FOR HELICAL PINION
C... *          GENERATED BY CONE CUTTER
C... *          AUTHOR: FAYDOR LITVIN
C... *          JIAO ZHANG
C... *
C... *
C... *****
C
C PURPOSE
C
C THIS PROGRAM IS USED TO FIND THE UDDERCUTTING CONDITIONS FOR A
C HELICAL PINION GENERATED BY CONE CUTTER
C
C NOTE
C
C THIS PROGRAM IS WRITTEN IN FORTRAN 77. IT CAN BE COMPILED BY V
C COMPILE IN IBM MAINFRAME OR FORTRAN COMPILER IN VAX SYSTEM.
C
C IMPLICIT REAL*8(A-H,O-Z)
C DOUBLE PRECISION KSIN,MU
C
C DEFINE PARAMETERS USED BY PROGRAMS
C
C (1) IN ANG LP ARE UNIT NUMBERS ASSIGNED TO THE INPUT AND OUTPUT
C DEVICES
C IN=5
C LP=6
C (2) NDBUG IS USE TO CONTROL THE AUXILIARY OUTPUT FOR DEBUGGING
C NDBUG=2
C (3) OTHER PARAMETERS(DON'T CHANGE)
C DR=DATAN(1.D0)/45.D0
C
C DEFINE INPUT PARAMTERS OF PROBLEM(USE INCH AS UNIT OF LENGTH)
C
C (1) PINION AND GEAR: PN=DIAMETRAL PITCH; N1=PINION TOOTH NUMBER;
C KSIN=PRESSURE ANGLE; BETAP=HELIX ANGEL OF PINION;
C HD=HEIGHT OF DEDENDUM OF PINION
C PN=10.D0
C N1=20
C KSIN=20.D0*DR
C BETAP=30.D0*DR
C HD=1.D0/PN
C (2) TOOL: ALPHA=HALF OF CONE VERTEX ANGLE(DEGREE);
C RC=RADIUS OF BOTTOM CIRCLE OF CONE;
C MU=TILT ANGLE OF MOUNTING TOOL
C ALPHA=20.D0*DR
C RC=0.35D0
C MUO=DATAN(DSIN(KSIN)*DTAN(BETAP))
C MU=0.*MUO
C (3) PROBLEM: NPROB=ID NO. OF PROBLEM (-1=GIVEN N1 AND HD, FIND IF
C UDDERCUTTING OCCUR; 0=GIVEN N1, FIND MAXIMUM HD WITHOUT UDDER-

```

```

C      CUTTING; 1=GIVEN HD, FIND MINIMUM N1 WITHOUT UNDERCUTTING);
C      N=NO. OF THETA VALUES USED CALCULATION;
C      DELTHE=INCREMENT OF THETA(DEGREE)
      NPROB= 1
      N=21
      DELTHE=1.D0*DR
C
C DESCRIPTION OF OUTPUT
C
C OUTPUT IS A STATEMENT BASED ON THE PROBLEM WITHOUT ANY LITERAL
C PARAMETER
C
C FIND AUXILIARY VALUES FOR CALCULATION
      SK=DSIN(KSIN)
      CK=DCOS(KSIN)
      SB=DSIN(BETAP)
      CB=DCOS(BETAP)
      SM=DSIN(MU)
      CM=DCOS(MU)
      SA=DSIN(ALPHA)
      CA=DCOS(ALPHA)
      PT=PN*CB
      RP=N1/2./PT
      CL=RC/SA
      CH=HD/CK/CM
      UU=CL-CH
      AA=CM*CB+SK*SM*SB
      BB=CA*CK*SB+SA*SK*CM*SB-SA*SM*CB
      CC=CA*CK*SB
      DD1=CA*SK-SA*CK*CM
      DD=DD1*CA
      EE1=SA*SK+CA*CK*CM
      EE=EE1*SA
      FF1=CK*SM
      FF=FF1*CA
      II=(CA*SK*CM*SB-SA*CK*SB-CA*SM*CB)*CA/SA
      WRITE (LP,90)
90  FORMAT(1H1)
      IF (NPROB) 5,15,25
C CHECK IF UNDERCUTTING OCCURS
5  WRITE (LP,10)
10  FORMAT (1H1/3X,'NO',7X,'THETA',12X,'A',13X,'B',14X,'C',10X,
+      'B**2-4.*A*C',3X,'U1/(RC/SIN(ALPHA))')
      UMIN=5.D0
      DO 45 I=1,N
      NN=(N+1)/2
      THE=DBLE(FLOAT(I-NN))*DELTHE
      ST=DSIN(THE)
      CT=DCOS(THE)
      GG=DD*CT+EE-FF*ST
      WW=(BB*BB+AA*AA)*EE+(BB*DD-AA*FF)*II+ST*ST*ST*(AA*AA*FF-BB*BB*FF
+      +2.*AA*BB*DD)-CT*CT*CT*(BB*BB*DD-AA*AA*DD+2.*AA*BB*FF)
+      -ST*(2.*AA*AA*FF+AA*BB*DD-AA*EE*II)+CT*(2.*BB*BB*DD+AA*BB*FF
+      +BB*EE*II)

```

```

XX=((AA*FF*SA*SA-CC*EE)*CT+(AA*DD*SA*SA-AA*EE*CA*CA)*ST+(AA*FF*CA
+      *CA-CC*DD))* (BB*CT+AA*ST+II)
YY=DD1*SA*CT-FF1*SA*ST-EE1*CA
ZZ=CM*CK
A=YY*WW
B=UU*(WW*ZZ+XX*YY)+RP*GG*GG*GG
C=UU*UU*XX*ZZ
D=B*B-4.*A*C
IF (D.GE.0.) THEN
U1=(-B-DSQRT(D))/2./A
U2=(-B+DSQRT(D))/2./A
U1=U1/CL
IF (UMIN.GT.U1) UMIN=U1
U2=U2/CL
THE=THE/DR
IF (NDEBUG.LT.1) WRITE (LP,100) I,THE,A,B,C,D,U1
100 FORMAT (1X,I4,8F15.7)
ELSE
IF (NDEBUG.LT.1) WRITE (LP,100) I,THE,A,B,C,DD
END IF
45 CONTINUE
IF (UMIN.LT.1.DO) WRITE (LP,110)
110 FORMAT (///1X,'UDDERCUTTING WILL OCCUR FOR YOUR DESIGN')
IF (UMIN.GE.1.DO) WRITE (LP,120)
120 FORMAT (///1X,'UDDERCUTTING WILL NOT OCCUR FOR YOUR DESIGN')
GO TO 35
C DETERMINE THE MAXIMUM ADDENDUM HEIGHT OF RACK CUTTER
15 WRITE (LP,20)
20 FORMAT (1H1/3X,'NO',7X,'THETA',12X,'A',13X,'B',14X,'C',10X,
+      'B**2-4.*A*C',3X,'ALLOWED RATIO OF HD/(1/PN)')
UMIN=5.DO
DO 55 I=1,N
THE=DBLE(FLOAT(I-NN))*DELTHE
ST=DSIN(THE)
CT=DCOS(THE)
GG=DD*CT+EE-FF*ST
WW=(BB*BB+AA*AA)*EE+(BB*DD-AA*FF)*II+ST*ST*ST*(AA*AA*FF-BB*BB*FF
+      +2.*AA*BB*DD)-CT*CT*CT*(BB*BB*DD-AA*AA*DD+2.*AA*BB*FF)
+      -ST*(2.*AA*AA*FF+AA*BB*DD-AA*EE*II)+CT*(2.*BB*BB*DD+AA*BB*FF
+      +BB*EE*II)
XX=((AA*FF*SA*SA-CC*EE)*CT+(AA*DD*SA*SA-AA*EE*CA*CA)*ST+(AA*FF*CA
+      *CA-CC*DD))* (BB*CT+AA*ST+II)
YY=DD1*SA*CT-FF1*SA*ST-EE1*CA
ZZ=CM*CK
A=XX*ZZ
B=CL*(WW*ZZ+XX*YY)
C=CL*CL*YY*WW+CL*RP*GG*GG*GG
D=B*B-4.*A*C
TEST=DABS(A/B)
EPS=1.D-16
THE=THE/DR
IF (D.GE.0.) THEN
IF (TEST.GT.EPS) THEN
U1=(-B+DSQRT(D))/2./A

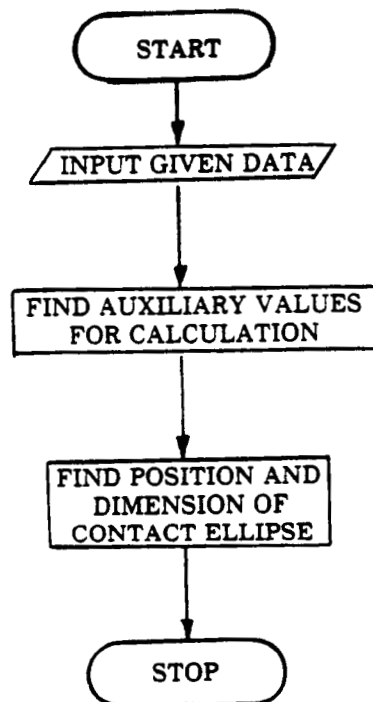
```

```

U2=(-B-DSQRT(D))/2./A
ELSE
U1=-C/B
U2=0.
END IF
U1=(CL-U1)*CK*PN
U2=(CL-U2)*CK*PN
IF (NDEBUG.LT.1) WRITE (LP,100) I,THE,A,B,C,D,U1,U2
ELSE
IF (NDEBUG.LT.1) WRITE (LP,100) I,THE,A,B,C,D
END IF
IF (UMIN.GT.U1) UMIN=U1
55 CONTINUE
WRITE (LP,200) UMIN
200 FORMAT (///1X,'TO AVOID UDDERCUTTING, IT IS NECESSARY TO KEEP DEDE
+NDUM OF PINION <=',F10.7,'/PN')
GO TO 35
C DETERMINE THE MINIMUM NO. OF TEETH FOR UNUNDERCUTTING
25 WRITE (LP,30)
30 FORMAT (1H1/3X,'NO',7X,'THETA',7X,'NO. OF TEETH')
RNMAX=0.
DO 65 I=1,N
THE=DBLE(FLOAT(I-NN))*DELTHE
ST=DSIN(THE)
CT=DCOS(THE)
GG=DD*CT+EE-FF*ST
WW=(BB*BB+AA*AA)*EE+(BB*DD-AA*FF)*II+ST*ST*ST*(AA*AA*FF-BB*BB*FF
+ 2.*AA*BB*DD)-CT*CT*CT*(BB*BB*DD-AA*AA*DD+2.*AA*BB*FF)
+ -ST*(2.*AA*AA*FF+AA*BB*DD-AA*EE*II)+CT*(2.*BB*BB*DD+AA*BB*FF
+ BB*EE*II)
XX=((AA*FF*SA*SA-CC*EE)*CT+(AA*DD*SA*SA-AA*EE*CA*CA)*ST+(AA*FF*CA
+ *CA-CC*DD))*(BB*CT+AA*ST+II)
YY=DD1*SA*CT-FF1*SA*ST-EE1*CA
ZZ=CM*CK
RR=-(CL*CL*YY*WW+CL*UU*(WW*ZZ+XX*YY)+UU*UU*XX*ZZ)/CL/GG**3
RN=2.*RR*PT
THE=THE/DR
IF (RNMAX.LT.RN) RNMAX=RN
IF (NDEBUG.LT.1) WRITE (LP,100) I,THE,RN
65 CONTINUE
WRITE (LP,300) RNMAX
300 FORMAT (/// 1X,'WITHOUT UDDERCUTTING, MINIMUM TOOTH NO. OF PINION
+IS:',F11.7)
35 WRITE (LP,400)
400 FORMAT(1H1)
STOP
END

```

### FLOWCHART FOR PROGRAM III



```

C... *****
C... *
C... *
C... *      PROGRAM III
C... * CONTACT ELLIPSIS FOR HELICAL PINION GENERATED BY
C... * CONE CUTTER IN MESHING WITH REGULAR HELICAL GEAR
C... *
C... *      AUTHORS: FAYDOR LITVIN
C... *      JIAO ZHANG
C... *
C... *****
C
C PURPOSE
C
C THIS PROGRAM IS USED TO FIND THE SHAPE AND ORIENTAION OF THE CONTACT
C ELLIPSE WHEN A HELICAL PINION CROWNED BY CONE CUTTER IS IN
C MESHING WITH A REGULAR HELICAL GEAR
C
C NOTE
C
C THIS PROGRAM IS WRITTEN IN FORTRAN 77. IT CAN BE COMPILED BY V
C COMPILER IN IBM MAINFRAME OR FORTRAN COMPILER IN VAX SYSTEM.
C
C IMPLICIT REAL*8(A-H,O-Z)
C DOUBLE PRECISION KSIN,MU,MUO,MUT,KSIC,KF,KH,KSP,KQP,KSG,KQG
C DIMENSION CMM(3,4),EFD(3),EHD(3),RND(3),RD(4),EFF(3),EHF(3),
C + RNF(3),RC(3),RF(3),R1(3)
C
C DEFINE PARAMETERS USED BY PROGRAMS
C
C (1) IN ANG LP ARE UNIT NUMBERS ASSIGNED TO THE INIUT AND OUTPUT
C DEVICES
C IN=5
C LP=6
C (2) NDBUG IS USE TO CONTROL THE AUXILIARY OUTPUT FOR DEBUGGING
C NDBUG=2
C (3) OTHER PARAMETERS(DON'T CHANGE)
C DR=DATAN(1.DO)/45.DO
C
C DEFINE INPUT PARAMTERS OF PROBLEM(USE INCH AS UNIT OF LENGTH)
C
C (1) PINION AND GEAR: PN=DIAMETRAL PITCH; N1=PINION TOOTH NUMBER;
C MPG=TOOTH NUMBER RATIO(GEAR TOOTH NO./N1);
C KSIN=PRESSURE ANGLE IN NORMAL SECTION;
C BETAP=HELIX ANGEL;
C HD=HEIGHT OF DEDENDUM OF PINION
C PN=10.DO
C N1=20
C MPG=2
C KSIN=20.DO*DR
C BETAP=10.DO*DR
C HD=1./PN
C (2) TOOL: ALPHA=HALF OF CONE VERTEX ANGLE(DEGREE);
C RC=RADIUS OF BOTTOM CIRCLE OF CONE;

```

```

C      MU=TILT ANGEL TO INSTALL PINION CUTTING TOOL
      ALPHA=80.D0*DR
      MUO=DATAN(DSIN(KSIN)*DTAN(BETAP))
      MU=0.*MUO
      RC=1.D0
C      (3) DEFORMATION: DEL=CONTACT DEFORMATION AT CONTACT POINT
      DEL=4.D-4
C      (4) OUTPUT: NU=NUMBER OF CONTACT POINTS IN MATING SURFACES FOR US
C      TO CALCULATE CONTACT ELLIPSES)
      NU=101
C
C      DESCRIPTION OF OUTPUT PARAMETER
C
C      R1=PINION RADIUS OF CONTACT POINT
C      ALPHA=THE ROTATION ANGLE BETWEEN PRINCIPAL DIRECTION OF PINION
C      TOOTH SURFACE AND AXES OF CONTACT ELLIPSE
C      A=LENGTH OF HALF SHORT AXIS OF CONTACT ELLIPSE
C      B=LENTH OF HALF LONG AXIS OF CONTACT ELLIPSE(ALONG DIRECTION OF
C      GEAR TOOTH LEHGTH)
C      RNF=UNIT NORMAL OF PINION TOOTH SURFACE AT CONTACT POINT
C      EFF=PRINCIPAL DIRECTION OF PINION TOOTH SURFACE AT CONTACT POINT
C      EHF=PRINCIPAL DIRECTION OF PINION TOOTH SURFACE AT CONTACT POINT
C
C      FIND AUXILIARY VALUES FOR CALCULATION
      SK=DSIN(KSIN)
      CK=DCOS(KSIN)
      SB=DSIN(BETAP)
      CB=DCOS(BETAP)
      SM=DSIN(MU)
      CM=DCOS(MU)
      SA=DSIN(ALPHA)
      CA=DCOS(ALPHA)
      KSIC=DATAN(SK/CK/CB)
      CKC=DCOS(KSIC)
      SKC=DSIN(KSIC)
      PT=PN*CB
      RP=N1/2./PT
      RB=RP*DCOS(KSIC)
      RA=RP+1./PN
      N2=N1*MPG
      RG=RP*MPG
      CL=RC/SA
      CH=HD/CK/CM
      A=CL-CH
      CMM(1,1)=CA*CK*CB+SA*SK*CM*CB+SA*SM*SB
      CMM(1,2)=SA*CK*CB-CA*SK*CM*CB-CA*SM*SB
      CMM(1,3)=SK*SM*CB-CM*SB
      CMM(1,4)=- (SK*CM*CB+SM*SB)
      CMM(2,1)=CA*SK-SA*CK*CM
      CMM(2,2)=SA*SK+CA*CK*CM
      CMM(2,3)=-CK*SM
      CMM(2,4)=CK*CM
      CMM(3,1)=CA*CK*SB+SA*SK*CM*SB-SA*SM*CB
      CMM(3,2)=SA*CK*SB-CA*SK*CM*SB+CA*SM*CB

```

```

CMM(3,3)=SK*SM*SB+CM*CB
CMM(3,4)=-SK*CM*SB+SM*CB
D1=CM*CB+SK*SM*SB
D2=SA*SM*CB-SB*(CA*CK+SA*SK*CM)
D3=CA*CK*SB
D4=CA*(CA*SK-SA*CK*CM)
D5=SA*(SA*SK+CA*CK*CM)
D6=CA*CK*SM
UL=CL-2.D0*CH
UU=CL
MUT=MUO/DR
WRITE (LP,110) RP,RA,RB,MUT
110 FORMAT (///1X,'RP=',F15.7,5X,'RA=',F15.7,5X,'RB=',F15.7,5X,
+          'MUO=',F15.7)
TH=0.D0
ST=DSIN(TH*DR)
CT=DCOS(TH*DR)
EFD(1)=ST
EFD(2)=0.D0
EFD(3)=-CT
EHD(1)=-CT*SA
EHD(2)=CA
EHD(3)=-ST*SA
RND(1)=CT*CA
RND(2)=SA
RND(3)=ST*CA
CALL MATMUL(CMM,EFD,EFF,3,3)
CALL MATMUL(CMM,EHD,EHF,3,3)
CALL MATMUL(CMM,RND,RNF,3,3)
WRITE (LP,10) TH,(EFF(M),M=1,3),(EHF(M),M=1,3),(RNF(M),M=1,3)
10 FORMAT(1H1,///3X,'THTEA:',F15.7,' DEGREE'///1X,'EFF:',3F15.7/1X,
+        'EHF:',3F15.7/1X,'NF :',3F15.7)
DO 15 J=1,NU
U=UU-(UU-UL)/FLOAT(NU-1)*FLOAT(J-1)
KF=-CA/SA/U
KH=0.D0
RD(1)=U*CT*SA
RD(2)=-U*CA
RD(3)=U*ST*SA
RD(4)=A
CALL MATMUL(CMM,RD,RC,3,4)
FEE=(U*(CT*D1+ST*D2)-A*((CT*CA*CA+SA*SA)*D1-ST*D3))/(CT*D4+D5-ST*
+ D6)/RP
RF(1)=RC(1)-RP*FEE
RF(2)=RC(2)+RP
RF(3)=RC(3)
RPP=DSQRT(RF(1)**2+RF(2)**2)
CALL PROT(RF,R1,FEE)
TEMP=RC(2)*EFF(1)-RF(1)*EFF(2)
B13=EHF(3)-KF*TEMP
B23=-EFF(3)
B33=-(RNF(1)*RF(1)+RNF(2)*RF(2)+KF*TEMP*TEMP)
SIGMA=0.5D0*DATAN(2.*B13*B23/(B23*B23-B13*B13-(KF-KH)*B33))
COE1=(B23*B23-B13*B13-(KF-KH)*B33)/B33/DCOS(2.*SIGMA)

```

```

COE2=KF+KH+(B13*B13+B23*B23)/B33
KSP=(COE2-COE1)/2.D0
KQP=(COE2+COE1)/2.D0
IF (NDEBUG.LT.1) WRITE (LP,20) U,FEE,RPP,(R1(M),M=1,3),
+      B13,B23,B33,SIGMA,KSP,KQP
20 FORMAT (//1X,'U  =',F15.7,5X,'FEE=',F15.7,5X,'R1  =',F15.7/
+      1X,'XP  =',F15.7,5X,'YP  =',F15.7,5X,'ZP  =',F15.7/
+      1X,'B13=',F15.7,5X,'B23=',F15.7,5X,'B33=',F15.7/
+      1X,'SIG=',F15.7,5X,'KSP=',F15.7,5X,'KQP=',F15.7)
W=-(A-U)
KSG=0.
KQG=1./(RG*SK*(CKC/CK/CB)**2+W*CM*CK/SK)
G1=KSP-KQP
G2=KSG-KQG
S1=KSP+KQP
S2=KSG+KQG
SIGMGP=-SIGMA
ALPHA1=0.5D0*DATAN(G2*DSIN(2.*SIGMGP)/(G1-G2*DCOS(2.*SIGMGP)))
ALPHA2=ALPHA1+SIGMGP
AA=(S1-S2-DSQRT(G1*G1-2.*G1*G2*DCOS(2.*SIGMGP)+G2*G2))/4.
BB=(S1-S2+DSQRT(G1*G1-2.*G1*G2*DCOS(2.*SIGMGP)+G2*G2))/4.
AAA=1./DSQRT(DABS(AA))
BBB=1./DSQRT(DABS(BB))
RATIO=BBB/AAA
ALPHA1=ALPHA1/DR
ALPHA2=ALPHA2/DR
C      WRITE (LP,130) G1,G2,S1,S2,ALPHA1
C 130 FORMAT (1X,5F15.7)
      WRITE (LP,30) RPP,ALPHA2,AAA,BBB,RATIO
30 FORMAT (1X,'R1  =',F15.7,5X,'ALP=',F15.7,5X,'A  =',F15.7,5X,
+      'B  =',F15.7,5X,'B/A=',F15.7)
      IF (RPP.GT.RA) GO TO 5
15 CONTINUE
5 STOP
END

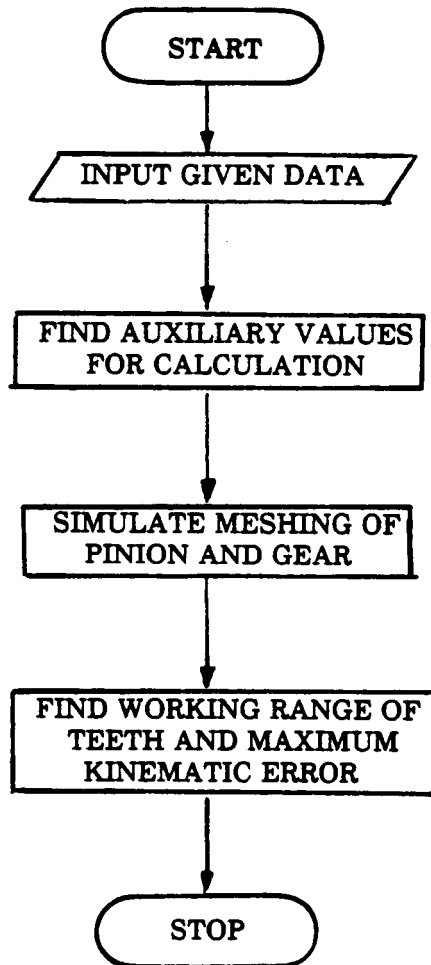
C
C
SUBROUTINE MATMUL(CMM,A,B,N,M)
C THIS SUBROUTINE IS USED TO MULTIPLY THE MATRIX CMM(N*M) BY THE MATRIX
C A(M*1). THE RESULT IS STORED IN THE MATRIX B(N*1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION CMM(3,M),A(M),B(N)
      DO 5 I=1,N
5 B(I)=0.
      DO 15 I=1,N
      DO 15 J=1,M
15 B(I)=B(I)+CMM(I,J)*A(J)
      RETURN
      END

C
C
SUBROUTINE PROT(A,B,FEE)
C THIS SUBROUTINE IS USED TO ROTATE COORDINATE SYSTEM PLANARLY IN XOY
C THROUGH ANGLE FEE

```

```
DOUBLE PRECISION A(3),B(3),FEE
B(1)=A(1)*DCOS(FEE)+A(2)*DSIN(FEE)
B(2)=A(2)*DCOS(FEE)-A(1)*DSIN(FEE)
B(3)=A(3)
RETURN
END
```

## FLOWCHART FOR PROGRAM IV



```

C... *****
C... *
C... *
C... *
C... *          PROGRAM IV
C... * TRANSMISSION ERRORS OF HELICAL PINION GENERATED
C... *          BY CUTTER WITH CONE OR REVOLUTE SURFACE
C... *          IN MESHING WITH REGULAR HELICAL GEAR
C... *
C... *          AUTHORS: FAYDOR LITVIN
C... *          JIAO ZHANG
C... *
C... *****
C
C PURPOSE
C
C THIS PROGRAM IS USED TO CALCULATE THE TRANSMISSION ERRORS OF A
C PINION GENERATED BY THE CUTTER WITH CONE OR REVOLUTE SURFACE
C IN MESHING WITH A MISALIGNED REGULAR HELICAL GEAR
C
C NOTE
C
C THIS PROGRAM IS WRITTEN IN FORTRAN 77. IT CAN BE COMPILED BY V
C COMPILER IN IBM MAINFRAME OR FORTRAN COMPILER IN VAX SYSTEM.
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      DOUBLE PRECISION KSIN,MU,MUO,KSIC
C      DIMENSION Z(99),ANGLE(99),ERROR(99),ERR(99)
C      COMMON /BLOCK1/ X(11),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
C      + EPSI,DELTA,NC,NE,NDIM
C      COMMON /BLOCK2/ S(4,4),CMM(4,4),CMCD(4,4),C,R,RP,RG,A1,HD,RL,KSIC,
C      + ALPHA,CK,SK,CB,SB,CM,SM,CA,SA,CKC,SKC,NT00L
C
C DEFINE PARAMETERS USED BY PROGRAMS
C
C (1) IN ANG LP ARE UNIT NUMBERS ASSIGNED TO THE INPUT AND OUTPUT
C DEVICES
C      IN=5
C      LP=6
C
C (2) NDBUG IS USE TO CONTROL THE AUXILIARY OUTPUT FOR DEBUGGING
C      NDBUG=1
C
C (3) NC IS THE UPPER LIMITATION OF REPEATATION FOR SOLVING NONLINEAR
C EQUATIONS;
C      EPSI IS THE CLEARANCE OF FUNCTION VALUES WHEN THE FUNCTIONS
C IS CONSIDERED AS SOLVED (ALL FUNCTIONS HAVE FORMS OF F(X)=0);
C      DELTA IS THE RELATIVE DIFFERENCE FOR TAKING DERIVATIVES
C      NC, EPSI AND DELTA MAY BE CHANGED WHEN SOLUTIONS ARE DIVERGENT
C OR LESS ACCURATE
C      NC=100
C      DELTA=1.D-3
C      EPSI=1.D-12
C
C (4) OTHER PARAMETERS(DON'T CHANGE)
C      NDIM=10
C      NE=5

```

```

DR=DATAN(1.D0)/45.D0
C  DEFINE INPUT PARAMTERS OF PROBLEM(USE INCH AS UNIT OF LENGTH)
C  (1) PINION AND GEAR: PN=DIAMETRAL PITCH; NP=PINION TOOTH NUMBER;
C      RMPG=TOOTH NUMBER RATIO(GEAR TOOTH NO./NP);
C      KSIN=PRESSURE ANGLE IN NORMAL SECTION;
C      BETAP=HELIX ANGLE OF PINION AND GEAR;
C      HD=HEIGHT OF DEDEDENDUM OF PINION;
C      COE=COEFF. OF CENTRAL DISTANCE(USUALLY COE=1.)
      PN=10.D0
      NP=20
      RMPG=2.D0
      KSIN=20.D0*DR
      BETAP=15.D0*DR
      HD=1.D0/PN
      COE=1.000D0
C  (2) TOOL: ALPHA=HALF OF CONE VERTEX ANGLE(DEGREE);
C      RC=RADIUS OF BOTTOM CIRCLE OF CONE
C      R=RADIUS OF ARC
C      MU=TILT ANGEL OF MOUNTING TOOL
C      NTOOL=TOOL ID NO. (1=CONE SURFACE, 2=REVOLUTE SURFACE)
      NTOOL=2
      ALPHA=20.D0*DR
      RC=10.3527D0
      R=3.0D1
      MUO=DATAN(DSIN(KSIN)*DTAN(BETAP))
      MU= 0.*MUO
C  (3) MISALIGNMENT: NMIS=ID NO.(1=CROSSING AXES, 2=INTERSECTING AXES);
C      NG=NO. OF MISALIGNED ANGLES TO BE SIMULATED (FROM -(NG-1)/2 TO
C      (NG-1)/2 TIMES GAMMAI WITH ODD NG);
C      GAMMAI=INCREMENT OF MISALIGNED ANGLE(MINUTE);
      NMIS=2
      NG=2
      GAMMAI=5.D0
C  (4) OUTPUT: FEEI=INCREMENT OF ROTATION ANGLE OF PINION(DEGREEE)
      FEEI=1.0D0*DR
C
C  DESCRIPTION OF OUTPUT PARAMERTERS
C
C      FEE1=ROTATION ANGLE OF PINION
C      FEE2=ROTATION ANGLE OF GEAR
C      RP=RADIUS OF PINION CONTACT POINT
C      RG=RADIUS OF GEAR CONTACT POINT
C
C  FIND AUXILIARY VALUES FOR CALCULATION
C
      SK=DSIN(KSIN)
      CK=DCOS(KSIN)
      SB=DSIN(BETAP)
      CB=DCOS(BETAP)
      SM=DSIN(MU)
      CM=DCOS(MU)
      SA=DSIN(ALPHA)
      CA=DCOS(ALPHA)

```

```

KSIC=DATAN(SK/CK/CB)
CKC=DCOS(KSIC)
SKC=DSIN(KSIC)
PT=PN*CB
RP=NP/2./PT
C RB=RP*DCOS(KSIC)
C RA=RP+1./PN
C NG=NP*RMPPG
RG=RP*RMPPG
RL=RC*CA/SA
C CH=HD/CK/CM
A1=RL/CA-HD/CK/CM
C=(RP+RG)*COE
NCOEF=360.D0*DR/FEEI/FLOAT(N1)+0.3
N=NCOEF*2+1
CALL INTMAT(CMCD,4,4)
AA=RL*SA*SA/CA-HD/CK/CM
CMCD(1,1)=CA*CK*CB+SA*SK*CM*CB+SA*SM*SB
CMCD(1,2)=SA*CK*CB-CA*SK*CM*CB-CA*SM*SB
CMCD(1,3)=SK*SM*CB-CM*SB
CMCD(1,4)=-RL*SA*CK*CB-AA*(SK*CM*CB+SM*SB)
CMCD(2,1)=CA*SK-SA*CK*CM
CMCD(2,2)=SA*SK+CA*CK*CM
CMCD(2,3)=-CK*SM
CMCD(2,4)=-RL*SA*SK+AA*CK*CM
CMCD(3,1)=CA*CK*SB+SA*SK*CM*SB-SA*SM*CB
CMCD(3,2)=SA*CK*SB-CA*SK*CM*SB+CA*SM*CB
CMCD(3,3)=SK*SM*SB+CM*CB
CMCD(3,4)=-RL*SA*CK*SB+AA*(-SK*CM*SB+SM*CB)
CALL INTMAT(S,4,4)
NGG=(NG-1)/2
DO 505 LL=1,NG
GAMMA=GAMMAI*FLOAT(LL-NGG)/60.D0*DR
CG=DCOS(GAMMA/60.*DR)
SG=DSIN(GAMMA/60.*DR)
IF (NMIS.EQ.1) THEN
WRITE (LP,500) COE,GAMMA
500 FORMAT (1H1,///,1X,'C=',F4.2,'*(RP+RG)      CROSSING ANGLE=',
+          F5.1,'(M)')
S(1,1)=CG
S(1,3)=-SG
S(3,1)=SG
S(3,3)=CG
ELSE
WRITE (LP,501) COE,GAMMA
501 FORMAT (1H1,///,1X,'C=',F4.2,'*(RP+RG)      INTERSECTING ANGLE=',
+          F5.1,'(M)')
S(2,2)=CG
S(2,3)=-SG
S(3,2)=SG
S(3,3)=CG
END IF
DO 205 L=1,2
DO 5 I=1,NE

```

```

5 X(I)=0.D0
  IF (NTOOL.EQ.1) X(5)=RL/CA-HD/CK/CM
C   WRITE (6,1100) (X(JK),JK=1,5),RL,CA,HD,CK,CM
C1100 FORMAT (1X,'#####',5F15.7/7X,5F15.7)
      DO 15 I=1,N
        X(7)=FEEI*FLOAT(I-(N+1)/2)
        IF (L.EQ.1) X(7)=0.D0
C       X(5)=DARSIN(RP*X(7)/SK/R)
        X(2)=X(7)/RMPG
        CALL NONLIN
        X(8)=X(2)+X(3)
        IF (L.EQ.1) THEN
          XIN=X(8)
          WRITE (LP,10)
10      FORMAT (///8X,'FEE1(D)',8X,'FEE2(D)',8X,'K-ERROR(S)',5X,
+             'RP',13X,'RG',F15.7/)
          GO TO 205
        END IF
        X(8)=X(8)-XIN
        X(7)=X(7)/DR
        X(8)=X(8)/DR
        X(9)=(X(8)-X(7)/RMPG)*3600.D0
        Z(I)=X(8)
        ERR(I)=X(9)
        WRITE (LP,20) (X(J),J=7,11)
20      FORMAT (1X,5F15.7,F15.7)
C       WRITE (LP,30) (X(J),J=1,6)
C       30      FORMAT (1X,'#####',6F15.7)
C       WRITE (LP,30) XP,YP,ZP,XG,YG
15      CONTINUE
      NT=N-NCOEF
      FII=FEEI/DR/2.D0*FLOAT(NCOEF)
      WRITE (LP,80)
80      FORMAT (//,' FIND THE WORKING RANGE FOR ONE TOOTH:',F15.7/)
      DO 55 I=1,NT
        X(7)=FEEI*FLOAT(I-(N+1)/2)/DR/2.D0
        X(8)=X(7)+FII
        KK=I+NCOEF
        ANGLE(I)=Z(KK)-Z(I)
        ERROR(I)=(ANGLE(I)-FII)*3600.D0
        WRITE (LP,60) X(7),X(8),ANGLE(I),ERROR(I)
60      FORMAT (1X,'( ',F7.2,'----',F7.2,'):',F15.7,F15.7)
55      CONTINUE
      DO 95 I=1,NT
        ATEMP2=ERROR(I)
        IF (I.NE.1) THEN
          IF (ATEMP1*ATEMP2.LE.0.D0) GOTO 105
        END IF
        ATEMP1=ATEMP2
95      CONTINUE
      WRITE (LP,160)
160     FORMAT (//1X,'MESHING IS DISCONTINUOUS')
105     IF (DABS(ATEMP1).LT.DABS(ATEMP2)) I=I-1
      EMAX=0.

```

```

    EMIN=0.
    DO 135 J=1,NCOEF
    KS=I+J-1
    ET=ERR(KS)
    IF (ET.LT.EMIN) EMIN=ET
    IF (ET.GT.EMAX) EMAX=ET
135 CONTINUE
    ET=EMAX-EMIN
    KK=I+NCOEF
    WRITE (LP,170) Z(I),Z(KK),ET
170 FORMAT (//1X,'WORKING RANGE FOR ONE TOOTH: ',F7.2,'----',F7.2/
+          1X,'THE MAXIMUM KINEMATIC ERROR: ',F15.7,' (S)',I2)
205 CONTINUE
505 CONTINUE
    STOP
    END

```

C

SUBROUTINE FUNC

C

```

    IMPLICIT REAL*8(A-H,O-Z)
    DOUBLE PRECISION KSIN,MU,MUO,KSIC
    COMMON /BLOCK1/ X(11),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
+          EPSI,DELTA,NC,NE,NDIM
    COMMON /BLOCK2/ S(4,4),CMM(4,4),CMCD(4,4),C,R,RP,RG,A1,HD,RL,KSIC,
+          ALPHA,CK,SK,CB,SB,CM,SM,CA,SA,CKC,SKC,NTOOL
    DIMENSION RD(4),RND(4),RC(4),RNC(4),RF(4)
    CFEE=DCOS(X(3))
    SFEE=DSIN(X(3))
    CT=DCOS(X(4))
    ST=DSIN(X(4))
    CF=DCOS(X(7))
    SF=DSIN(X(7))
    IF (NTOOL.EQ.1) THEN
    RD(1)=X(5)*CT*SA
    RD(2)=RL-X(5)*CA
    RD(3)=X(5)*ST*SA
    RND(1)=CT*CA
    RND(2)=SA
    RND(3)=ST*CA
    ELSE
    CAL=DCOS(ALPHA+X(5))
    SAL=DSIN(ALPHA+X(5))
    SL2=DSIN(X(5)/2.)
    CAL2=DCOS(ALPHA+X(5)/2.)
    SAL2=DSIN(ALPHA+X(5)/2.)
    RD(1)=(A1*SA-2.*R*SL2*SAL2)*CT
    RD(2)=HD/CK/CM*CA+2.*R*SL2*CAL2
    RD(3)=(A1*SA-2.*R*SL2*SAL2)*ST
    RND(1)=CAL*CT
    RND(2)=SAL
    RND(3)=CAL*ST
    END IF
    RD(4)=1.
    CALL MATMUL(CMCD,RD,RC,4,4)

```

```

CALL MATMUL (CMCD, RND, RNC, 3, 3)
X(6) = (RC(1) - RC(2) * RNC(1) / RNC(2)) / RP
RF(1) = RC(1) - RP * X(6)
RF(2) = RC(2) + RP
RF(3) = RC(3)
RF(4) = 1.
CFPF1 = DCOS(X(6) + X(7))
SFPP1 = DSIN(X(6) + X(7))
XPF = RF(1) * CFPF1 + RF(2) * SFPP1
YPP = RF(2) * CFPF1 - RF(1) * SFPP1
ZPF = RF(3)
XNPF = RNC(1) * CFPF1 + RNC(2) * SFPP1
YNPF = RNC(2) * CFPF1 - RNC(1) * SFPP1
ZNPF = RNC(3)
CF2FG = DCOS(X(3))
SF2FG = DSIN(X(3))
A2 = -X(1) * SB + RG * X(2)
RFG1 = CK * CK / CKC * DCOS(X(3) + KSIC) * A2 + RG * SF2FG
RFG2 = CK * CK / CKC * DSIN(X(3) + KSIC) * A2 - RG * CF2FG
RFG3 = X(1) * (CB + SK * SK * SB * SB / CB) - RG * X(2) * SK * SK * SB / CB
RNFG1 = CK * CB * CF2FG - SK * SF2FG
RNFG2 = CK * CB * SF2FG + SK * CF2FG
RNFG3 = CK * SB
XGF = RFG1 * S(1, 1) + RFG2 * S(1, 2) + RFG3 * S(1, 3)
YGF = RFG1 * S(2, 1) + RFG2 * S(2, 2) + RFG3 * S(2, 3)
ZGF = RFG1 * S(3, 1) + RFG2 * S(3, 2) + RFG3 * S(3, 3)
XNGF = RNFG1 * S(1, 1) + RNFG2 * S(1, 2) + RNFG3 * S(1, 3)
YNGF = RNFG1 * S(2, 1) + RNFG2 * S(2, 2) + RNFG3 * S(2, 3)
ZNGF = RNFG1 * S(3, 1) + RNFG2 * S(3, 2) + RNFG3 * S(3, 3)
C   WRITE (6, 100) X(1), X(2), X(3), X(4), X(5)
C   WRITE (6, 100) XPF, YPF, ZPF, XGF, YGF, ZGF
C 100 FORMAT (1X, '%%%%', 8F15.7)
C   WRITE (6, 100) XNPF, YNPF, ZNPF, XNGF, YNGF, ZNGF
C   Y(1) = XPF - XGF
C   Y(2) = YPF - YGF - C
C   Y(3) = ZPF - ZGF
C   Y(4) = YNPF - YNGF
C   Y(5) = ZNPF - ZNGF
C   X(10) = DSQRT(XPF * XPF + YPF * YPF)
C   X(11) = DSQRT(XGF * XGF + YGF * YGF)
C   WRITE (6, 20) (Y(II), II=1, 5)
C 20  FORMAT (1X, '$$$$ ', 6F15.7)
C   RETURN
C   END

C
C   SUBROUTINE INTMAT (A, N, M)
C   THIS SUBROUTINE IS USED TO INITIATE THE MATRIX, WITH UNIT DIAGONAL
C   ELEMENTS AND NULL OTHER ELEMENTS
C   IMPLICIT REAL *8 (A-H, O-Z)
C   DIMENSION A(4, 4)
C   DO 5 I=1, N
C   DO 5 J=1, M
C   A(I, J) = 0.
C   IF (I.EQ.J) A(I, J) = 1.

```

```

5 CONTINUE
  RETURN
  END

C
  SUBROUTINE MATMUL(CMM,A,B,N,M)
C THIS SUBROUTINE IS USED TO MULTIPLY THE MATRIX CMCD(N*M) BY THE MATRIX
C A(M*1). THE RESULT IS STORED IN THE MATRIX B(N*1)
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION CMM(4,4),A(4),B(4)
  DO 5 I=1,N
5 B(I)=0.
  DO 15 I=1,N
  DO 15 J=1,M
15 B(I)=B(I)+CMM(I,J)*A(J)
  RETURN
  END

C
C
  SUBROUTINE NONLIN
C
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /BLOCK1/ X(11),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
+ EPSI,DELTA,NC,NE,NDIM
C
  DO 5 I=1,NC
  CALL FUNC
C WRITE (6,10) I,(X(J),Y(J),J=1,5)
C 10 FORMAT(1X,'***',15/5(1X,2D15.7/))
  DO 15 J=1,NE
  IF (DABS(Y(J)).GT.EPSI) GO TO 25
15 CONTINUE
  GO TO 105
25 DO 35 J=1,NE
35 Y1(J)=Y(J)
  DO 45 J=1,NE
  DIFF=DABS(X(J))*DELTA
  IF (X(J).EQ.0.D0) DIFF=DELTA
  XMAM=X(J)
  X(J)=X(J)-DIFF
  CALL FUNC
  X(J)=XMAM
  DO 55 K=1,NE
55 A(K,J)=(Y1(K)-Y(K))/DIFF
45 CONTINUE
  DO 65 J=1,NE
65 Y(J)=-Y1(J)
  CALL DECOMP (NDIM,NE,A,COND,IPVT,WORK)
  CALL SOLVE (NDIM,NE,A,Y,IPVT)
  DO 75 J=1,NE
75 X(J)=X(J)+Y(J)
  5 CONTINUE
105 RETURN
  END

C

```

```

C
C
C      SUBROUTINE DECOMP (NDIM,N,A,COND,IPVT,WORK)
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      DIMENSION A(NDIM,N),WORK(N),IPVT(N)
C
C      DECOMPOSES AREAL MATRIX BY GAUSSIAN ELIMINATION,
C      AND ESTIMATES THE CONDITION OF THE MATRIX.
C
C      -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
C      M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
C
C      USE SUBROUTINE SOLVE TO COMPUTE SOLUTIONS TO LINEAR SYSTEM.
C
C      INPUT..
C
C      NDIM = DECLARED ROW DIMENSION OF THE ARRAY CONTAINING  A
C      N     = ORDER OF THE MATRIX
C      A     = MATRIX TO BE TRIANGULARIZED
C
C      OUTPUT..
C
C      A  CONTAINS AN UPPER TRIANGULAR MATRIX  U  AND A PREMUTED
C      VERSION OF A LOWER TRIANGULAR MATRIX  I-L  SO THAT
C      (PERMUTATION MATRIX)*A=L*U
C
C      COND = AN ESTIMATE OF THE CONDITION OF  A.
C      FOR THE LINEAR SYSTEM  A*X = B , CHANGES IN  A  AND  B
C      MAY CAUSE CHANGES  COND  TIMES AS LARGE IN  X.
C      IF  COND+1.0 .EQ. COND , A IS SINGULAR TO WORKING
C      PRECISION.  COND  IS SET TO  1.0D+32  IF EXACT
C      SINGULARITY IS DETECTED.
C
C      IPVT      = THE PIVOT VECTOR
C      IPVT(K)   = THE INDEX OF THE K-TH PIVOT ROW
C      IPVT(N)   = (-1)**(NUMBER OF INTERCHANGES)
C
C      WORK SPACE..  THE VECTOR  WORK  MUST BE DECLARED AND INCLUDED
C      IN THE CALL.  ITS INPUT CONTENTS ARE IGNORED.
C      ITS OUTPUT CONTENTS ARE USUALLY UNIMPORTANT.
C
C      THE DETERMINANT OF  A  CAN BE OBTAINED ON OUTPUT BY
C      DET(A) = IPVT(N) * A(1,1) * A(2,2) * ... * A(N,N) .
C
C      IPVT(N)=1
C      IF (N.EQ.1) GO TO 150
C      NM1=N-1
C
C                                     COMPUTE THE 1-NORM OF  A .
C
C      ANORM=0.D0
C      DO 20 J=1,N
C        T=0.D0
C        DO 10 I=1,N
10      T=T+DABS(A(I,J))

```

```

      IF (T.GT.ANORM) ANORM=T
20  CONTINUE
C          DO GAUSSIAN ELIMINATION WITH PARTIAL
C          PIVOTING.
      DO 70 K=1,NM1
      KP1=K+1
C          FIND THE PIVOT.
      M=K
      DO 30 I=KP1,N
      IF (DABS(A(I,K)).GT.DABS(A(M,K))) M=I
30  CONTINUE
      IPVT(K)=M
      IF (M.NE.K) IPVT(N)=-IPVT(N)
      T=A(M,K)
      A(M,K)=A(K,K)
      A(K,K)=T
C          SKIP THE ELIMINATION STEP IF PIVOT IS ZERO.
      IF (T.EQ.0.D0) GO TO 70
C
C          COMPUTE THE MULTIPLIERS.
      DO 40 I=KP1,N
40  A(I,K)=-A(I,K)/T
C          INTERCHANGE AND ELIMINATE BY COLUMNS.
      DO 60 J=KP1,N
      T=A(M,J)
      A(M,J)=A(K,J)
      A(K,J)=T
      IF (T.EQ.0.D0) GO TO 60
      DO 50 I=KP1,N
50  A(I,J)=A(I,J)+A(I,K)*T
60  CONTINUE
70  CONTINUE
C
C  COND = (1-NORM OF A)*(AN ESTIMATE OF THE 1-NORM OF A-INVERSE)
C  THE ESTIMATE IS OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
C  SMALL SINGULAR VECTOR. THIS INVOLVES SOLVING TWO SYSTEMS
C  OF EQUATIONS, (A-TRANPOSE)*Y = E AND A*Z = Y WHERE E
C  IS A VECTOR OF +1 OR -1 COMPONENTS CHOSEN TO CAUSS GROWTH IN Y.
C  ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)
C
C          SOLVE (A-TRANPOSE)*Y = E .
      DO 100 K=1,N
      T=0.D0
      IF (K.EQ.1) GO TO 90
      KM1=K-1
      DO 80 I=1,KM1
80  T=T+A(I,K)*WORK(I)
90  EK=1.D0
      IF (T.LT.0.D0) EK=-1.D0
      IF (A(K,K).EQ.0.D0) GO TO 160
100 WORK(K)=- (EK+T)/A(K,K)
      DO 120 KB=1,NM1
      K=N-KB
      T=0.D0

```

```

        KP1=K+1
        DO 110 I=KP1,N
110     T=T+A(I,K)*WORK(K)
        WORK(K)=T
        M=IPVT(K)
        IF (M.EQ.K) GO TO 120
        T=WORK(M)
        WORK(M)=WORK(K)
        WORK(K)=T
120 CONTINUE
C
        YNORM=0.DO
        DO 130 I=1,N
130     YNORM=YNORM+DABS(WORK(I))
C
C                                     SOLVE  $A*Z = Y$ 
        CALL SOLVE (NDIM,N,A,WORK,IPVT)
C
        ZNORM=0.DO
        DO 140 I=1,N
140     ZNORM=ZNORM+DABS(WORK(I))
C
C                                     ESTIMATE THE CONDITION.
        COND=ANORM*ZNORM/YNORM
        IF (COND.LT.1.DO) COND=1.DO
        RETURN
C
C                                     1-BY-1 CASE..
150 COND=1.DO
        IF (A(1,1).NE.0.DO) RETURN
C
C                                     EXACT SINGULARITY
160 COND=1.0D32
        RETURN
        END
        SUBROUTINE SOLVE (NDIM,N,A,B,IPVT)
C
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION A(NDIM,N),B(N),IPVT(N)
C
C     SOLVES A LINEAR SYSTEM,  $A*X = B$ 
C     DO NOT SOLVE THE SYSTEM IF DECOMP HAS DETECTED SINGULARITY.
C
C     -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
C       M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
C
C     INPUT..
C
C       NDIM = DECLARED ROW DIMENSION OF ARRAY CONTAINING A
C       N     = ORDER OF MATRIX
C       A     = TRIANGULARIZED MATRIX OBTAINED FROM SUBROUTINE DECOMP
C       B     = RIGHT HAND SIDE VECTOR
C       IPVT  = PIVOT VECTOR OBTAINED FROM DECOMP
C
C     OUTPUT..

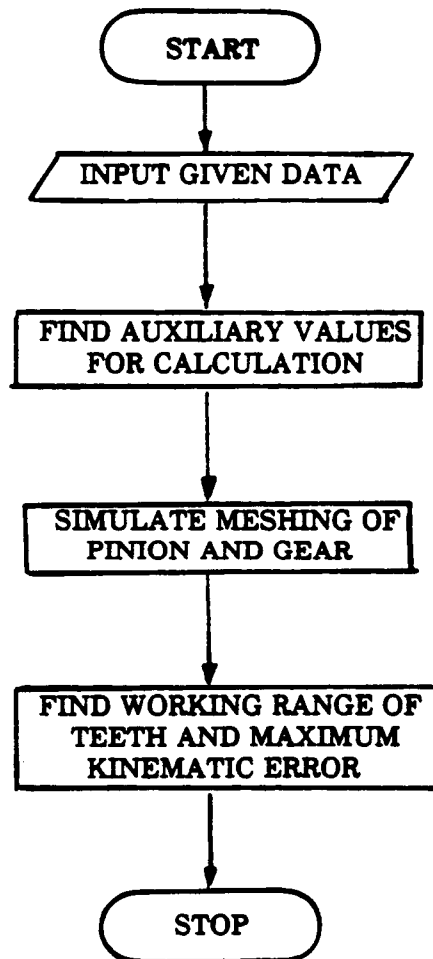
```

```

C
C      B = SOLUTION VECTOR,  X
C
C      DO THE FORWARD ELIMINATION.
      IF (N.EQ.1) GO TO 50
      NM1=N-1
      DO 20 K=1,NM1
        KP1=K+1
        M=IPVT(K)
        T=B(M)
        B(M)=B(K)
        B(K)=T
        DO 10 I=KP1,N
10      B(I)=B(I)+A(I,K)*T
20 CONTINUE
C      NOW DO THE BACK SUBSTITUTION.
      DO 40 KB=1,NM1
        KM1=N-KB
        K=KM1+1
        B(K)=B(K)/A(K,K)
        T=-B(K)
        DO 30 I=1,KM1
30      B(I)=B(I)+A(I,K)*T
40 CONTINUE
50 B(1)=B(1)/A(1,1)
      RETURN
      END
C
C

```

## FLOWCHART FOR PROGRAM V



```

C... *****
C... *
C... *
C... *
C... *          PROGRAM V
C... *  TRANSMISSION ERRORS OF CROWNED HELICAL PINION IN
C... *  MESHING WITH REGULAR HELICAL GEAR WITH PREDESIGNED
C... *  TRANSMISSION ERRORS AND CONTACT PATH
C... *
C... *          AUTHORS: FAYDOR LITVIN
C... *          JIAO ZHANG
C... *
C... *****

```

```

C
C PURPOSE
C

```

```

C THIS PROGRAM IS USED TO CALCULATE THE TRANSMISSION ERRORS OF A
C CROWNED HELICAL PINION AND A HELICAL GEAR WHEN THEIR AXES ARE
C MOUNTED WITH SOME ERRORS. THE MAGNITUDE OF TRANSMISSION ERRORS
C AND CONTACT PATH ARE PRE-DESIGNED.
C

```

```

C NOTE
C

```

```

C THIS PROGRAM IS WRITTEN IN FORTRAN 77. IT CAN BE COMPILED BY V
C COMPILER IN IBM MAINFRAME OR FORTRAN COMPILER IN VAX SYSTEM.
C

```

```

      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION KSIN,MU,KK,KSIC
      DIMENSION Z(99),ANGLE(99),ERROR(99),ERR(99),W(99),RPR(99),RGR(99),
+        ZP(99),ZG(99)
      COMMON /BLOCK1/ X(13),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
+        EPSI,DELTA,NC,NE,NDIM
      COMMON /BLOCK2/ S(4,4),AT(5),C,RP,RG,CK,SK,CB,SB,CKC,KSIC,TB,
+        COEG1,COEG2,RMGP,AA,DR,KK,R

```

```

C
C DEFINE PARAMETERS USED BY PROGRAMS
C

```

```

C (1) IN ANG LP ARE UNIT NUMBERS ASSIGNED TO THE INPUT AND OUTPUT
C DEVICES

```

```

      IN=5

```

```

      LP=6

```

```

C (2) NDBUG IS USE TO CONTROL THE AUXILIARY OUTPUT FOR DEBUGGING
      NDBUG=1

```

```

C (3) NC IS THE UPPER LIMITATION OF REPEATATION FOR SOLVING NONLINEAR
C EQUATIONS;

```

```

C EPSI IS THE CLEARANCE OF FUNCTION VALUES WHEN THE FUNCTIONS
C IS CONSIDERED AS SOLVED (ALL FUNCTIONS HAVE FORMS OF F(X)=0);
C DELTA IS THE RELATIVE DIFFERENCE FOR TAKING DERIVATIVES
C NC, EPSI AND DELTA MAY BE CHANGED WHEN SOLUTIONS ARE DIVERGENT
C OR LESS ACCURATE

```

```

      NC=100

```

```

      DELTA=1.D-3

```

```

      EPSI=1.D-10

```

```

C  (4) OTHER PARAMETERS (DON'T CHANGE)
      NDIM=10
      NE=5
      DR=DATAN(1.D0)/45.D0
C  DEFINE INPUT PARAMETERS OF PROBLEM (USE INCH AS UNIT OF LENGTH)
C  (1) PINION AND GEAR: PN=DIAMETRAL PITCH; NP=PINION TOOTH NUMBER;
      RMPG=TOOTH NUMBER RATIO (GEAR TOOTH NO./NP);
      KSIN=PRESSURE ANGLE IN NORMAL SECTION;
      BETAP=HELIX ANGLE OF PINION AND GEAR;
      COE=COEFF. OF CENTRAL DISTANCE (USUALLY COE=1.)
      PN=2.D0
      NP=12
      RMPG=94./12.
      KSIN=30.D0*DR
      BETAP=15.D0*DR
      COE=1.000D0
C  (2) PREDESIGN TRANSMISSION ERRORS AND CONTACT PATH:
      AA=LEVEL OF PREDESIGNED PARABOLIC TRANSMISSION ERRORS IN SECOND
      KK=COEFFICIENT OF CONTACT PATH DIRECTION (1.D-3=CROSS TOOTH
      SURFACE; 5.D6=ALONG TOOTH SURFACE)
      R=RADIUS OF ARC ATTACHED TO CHOSEN CONTACT PATH
      AA=25.D0
      KK=5.D6
      R=0.3584D0
C  (3) MISALIGNMENT: NMIS=ID NO. (1=CROSSING AXES, 2=INTERSECTING AXES);
      NG=NO. OF MISALIGNED ANGLES TO BE SIMULATED (FROM -(NG-1)/2 TO
      (NG-1)/2 TIMES GAMMAI WITH ODD NG);
      GAMMAI=INCREMENT OF MISALIGNED ANGLE (MINUTE);
      NMIS=1
      NG=3
      GAMMAI=3.D0
C  (4) OUTPUT: FEEI=INCREMENT OF ROTATION ANGLE OF PINION (DEGREEE)
      FEEI=1.0D0*DR
C
C  DESCRIPTION OF OUTPUT PARAMETERS
C
      FEE1=ROTATION ANGLE OF PINION
      FEE2=ROTATION ANGLE OF GEAR
      RP=RADIUS OF PINION CONTACT POINT
      RG=RADIUS OF GEAR CONTACT POINT
      ZP=LENGTH OF PINION CONTACT POINT FROM MIDDLE SECTION
      ZG=LENGTH OF GEAR CONTACT POINT FROM MIDDLE SECTION
C
C  FIND AUXILIARY VALUES FOR CALCULATION
C
C  DEFINE USEFUL CONSTANTS AND PARAMETERS FOR PINION AND GEAR
      RMGP=1./RMPG
      NG=NP*RMGP+0.5
      AA=AA*(2/3600.*DR*(NP/DR/180.)**2)
      SK=DSIN(KSIN)
      CK=DCOS(KSIN)
      SB=DSIN(BETAP)
      CB=DCOS(BETAP)
      TB=SB/CB

```

```

KSIC=DATAN(SK/CK/CB)
CKC=DCOS(KSIC)
SKC=DSIN(KSIC)
PT=PN*CB
RP=NP/2./PT
RPB=RP*CKC
RPA=RP+1.0/PN
RG=NG/2./PT
RGB=RG*CKC
RGA=RG+1.0/PN
WRITE (LP,56) RP,RPB,RPA,RG,RGB,RGA,AA
56 FORMAT (1X,'&&&&&&', 'RP=',F15.7,5X,'RPB=',F15.7,5X,'RPA=',F15.7/
+      1X,'&&&&&&', 'RG=',F15.7,5X,'RGB=',F15.7,5X,'RGA=',F15.7)
CON1=(1.+SK*SK*TB*TB)
AT(1)=(CK**4/CKC**2*KK**2-SK**4*TB**2)*RG*RG
AT(2)=(2.*SK*SK*SB*CON1-2.*KK**2*SB*CK**4/CKC**2)*RG
AT(3)=(-2.*CK*CK/CKC*SKC*KK**2-2.*KK*SK*SK*TB)*RG*RG
AT(4)=(2.*SB*CK*CK/CKC*SKC*KK**2+2.*KK*CB*CON1)*RG
AT(5)=CK**4/CKC**2*SB**2*KK**2-CB**2*CON1*CON1
C=(RP+RG)*COE
CALL INTMAT(S,4,4)
NGG=(NG-1)/2
DO 505 LL=1,NG
GAMMA=GAMMAI*FLOAT(LL-NGG)
CG=DCOS(GAMMA/60.*DR)
SG=DSIN(GAMMA/60.*DR)
IF (NMIS.EQ.1) THEN
WRITE (LP,500) COE,GAMMA
500 FORMAT (1H1,///,1X,'C=',F9.4,'*(RP+RG)      CROSSING ANGLE=',
+      F8.4,'(M)')
S(1,1)=CG
S(1,3)=-SG
S(3,1)=SG
S(3,3)=CG
ELSE
WRITE (LP,501) COE,GAMMA
501 FORMAT (1H1,///,1X,'C=',F4.2,'*(RP+RG)      INTERSECTING ANGLE=',
+      F5.1,'(M)')
S(2,2)=CG
S(2,3)=-SG
S(3,2)=SG
S(3,3)=CG
END IF
NCOEF=IDINT(360.*DR/FEEI/FLOAT(NP)+0.5)
N=2*NCOEF
NHALF=(N+1)/2
XIN=0.
DO 205 L=1,2
LSGN=(-1)**L
DO 15 I=1,NHALF
LI=NHALF+LSGN*(I-1)
IF (L.EQ.1) NMIN=LI
IF (L.EQ.2) NMAX=LI
X(7)=LSGN*FEEI*FLOAT(I-1)

```

```

      X(4)=X(7)
      X(2)=X(7)/RMPG
      X(3)=0.
      X(5)=(-(AT(2)*X(2)+AT(4))+DSQRT((AT(2)*X(2)+AT(4))**2-4.*AT(5)
+      *(AT(3)*X(2)+AT(1)*X(2)*X(2))))/2./AT(5)
      X(1)=0.
      CALL NONLIN
      X(8)=X(1)+X(2)
C   FIND INITIAL VALUE OF X(8)
      IF (L.EQ.1.AND.I.EQ.1) THEN
        XIN=X(8)
        WRITE (LP,51) XIN
51   FORMAT(1X,'XIN=',D15.7)
        GO TO 15
      END IF
C
      X(8)=X(8)-XIN
      W(LI)=X(7)/DR
      Z(LI)=X(8)/DR
      ERR(LI)=(X(8)-X(7)/RMPG)*3600.DO/DR
      RPR(LI)=X(10)
      RGR(LI)=X(11)
      ZP(LI)=X(12)
      ZG(LI)=X(13)
C   WRITE (LP,20) W(LI),Z(LI),ERR(LI),RPR(LI),RGR(LI)
15  CONTINUE
205 CONTINUE
      WRITE (LP,10)
10  FORMAT (///8X,'FEE1(D)',8X,'FEE2(D)',8X,'K-ERROR(S)',5X,
+          'RP',13X,'ZP',13X,'RG',13X,'ZG',F15.7/)
      DO 25 I=NMIN,NMAX
        WRITE (LP,20) W(I),Z(I),ERR(I),RPR(I),ZP(I),RGR(I),ZG(I)
20  FORMAT (1X,2F15.7,F12.4,4F15.7)
25  CONTINUE
      NT=NMAX-NCOEF
      FII=FEEI/DR/RMPG*FLOAT(NCOEF)
      WRITE (LP,80)
80  FORMAT (//,' FIND THE WORKING RANGE FOR ONE TOOTH:',F15.7/)
      DO 55 I=NMIN,NT
        X(7)=FEEI*FLOAT(I-(N+1)/2)/RMPG/DR
        X(8)=X(7)+FII
        KK=I+NCOEF
        ANGLE(I)=Z(KK)-Z(I)
        ERROR(I)=(ANGLE(I)-FII)*3600.DO
C   WRITE (LP,60) X(7),X(8),ANGLE(I),ERROR(I)
C 60  FORMAT (1X,'(',F7.2,'----',F7.2,'): ',F15.7,F15.7)
55  CONTINUE
      DO 95 I=NMIN,NT
        ATEMP2=ERROR(I)
        IF (I.NE.NMIN) THEN
          IF (ATEMP1*ATEMP2.LE.0.DO) GOTO 105
        END IF
        ATEMP1=ATEMP2
95  CONTINUE

```

```

        WRITE (LP,160)
160  FORMAT (//1X,'MESHING IS DISCONTINUOUS')
        GO TO 505
105  IF (DABS(ATEMP1).LT.DABS(ATEMP2)) I=I-1
        EMAX=0.
        EMIN=0.
        NTEMP=NCOEF+1
        DO 135 J=1,NTEMP
        KS=I+J-1
        ET=ERR(KS)
        IF (ET.LT.EMIN) EMIN=ET
        IF (ET.GT.EMAX) EMAX=ET
135  CONTINUE
        ET=EMAX-EMIN
        KK=I+NCOEF
        WRITE (LP,170) Z(I),Z(KK),ET
170  FORMAT (//1X,'WORKING RANGE FOR ONE TOOTH: ',F7.2,'----',F7.2/
+          1X,'THE MAXIMUM KINEMATIC ERROR: ',F12.4,' (S)',I2)
505  CONTINUE
        STOP
        END

C
        SUBROUTINE FUNC
C
        IMPLICIT REAL*8(A-H,O-Z)
        DOUBLE PRECISION KSIN,MU,KK,KSIC
        COMMON /BLOCK1/ X(13),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
+          EPSI,DELTA,NC,NE,NDIM
        COMMON /BLOCK2/ S(4,4),AT(5),C,RP,RG,CK,SK,CB,SB,CKC,KSIC,TB,
+          COEG1,COEG2,RMGP,AA,DR,KK,R
C
        *****
20  TEMP=CKC*(1.-AA*X(4)/(RMGP+1.))
        IF (DABS(TEMP).GT.1.D0) THEN
        X(4)=X(7)
        GOTO 20
        END IF
C
        WRITE (6,220) TEMP,X(4),AA
C 220  FORMAT(1X,'TEMP=',E15.7,'X(4)=',E15.7,'AA=',F15.7)
        RLAM=KSIC-DARCOS(TEMP)
        DLAMDF=-CKC*AA/(RMGP+1.)/DSQRT(1.-TEMP*TEMP)
        FEEGP=X(4)*RMGP-AA/2.*X(4)*X(4)+RLAM
        DFGDFP=RMGP-AA*X(4)+DLAMDF
        CONA=AT(5)
        CONB=AT(2)*FEEGP+AT(4)
        CONC=AT(3)*FEEGP+AT(1)*FEEGP*FEEGP
        COND=CONB*CONB-4.*CONA*CONC
        IF (COND.LT.0.) THEN
        X(4)=X(7)
        GOTO 20
        END IF
        COND=DSQRT(COND)
        TP=(-CONB+COND)/2./CONA
        IF (COND.EQ.0.0) THEN
        DTPDFG=-AT(2)/2./CONA

```

```

ELSE
  DTPDFG=(-AT(2)-(AT(2)*CONB-2.*CONA*(AT(3)+2.*AT(1)*FEEGP))/COND)
+      /2./CONA
END IF
C      *****
C X(4)=FEEP; X(7)=FEE1
  AP1=X(4)-X(7)
  AP2=AP1-RLAM
  AP3=AP2+KSIC
  AP4=X(4)-RLAM+KSIC
  DAP4DF=1.-DLAMDF
  CAP4=DCOS(AP4)
  SAP4=DSIN(AP4)
  AP=-TP*SB+RG*FEEGP
  DAPDF=(-SB*DTPDFG+RG)*DFGDFP
  MU=X(4)+KSIC-RLAM
  DMUDFP=1.+DLAMDF
  ALP1=R*(DSIN(MU+X(3))-DSIN(MU))
  ALP2=R*(DCOS(MU+X(3))-DCOS(MU))
  XPF=CK*CK/CKC*DCOS(AP3)*AP+RG*DSIN(AP2)-(RP+RG)*DSIN(AP1)+ALP2
  YPF=CK*CK/CKC*DSIN(AP3)*AP-RG*DCOS(AP2)+(RP+RG)*DCOS(AP1)+ALP1
  ZPF=TP*CB*CK*CK/CKC/CKC-SK*SK*TB*RG*FEEGP
C      *****
  DXDA=-DSIN(MU+X(3))
  DYDA=DCOS(MU+X(3))
  DXDF=CK*CK/CKC*(CAP4*DAPDF-SAP4*AP*DAP4DF)+RG*DCOS(X(4)-RLAM)
+      *(1.-DLAMDF)-(RP+RG)*DCOS(X(4))-ALP1*DMUDFP
  DYDF=CK*CK/CKC*(SAP4*DAPDF+CAP4*AP*DAP4DF)+RG*DSIN(X(4)-RLAM)
+      *(1.-DLAMDF)-(RP+RG)*DSIN(X(4))+ALP2*DMUDFP
  DZDF=CB*CK*CK/CKC/CKC*DTPDFG*DFGDFP-SK*SK*TB*RG*DFGDFP
  XNP=DYDA*DZDF
  YNP=-DXDA*DZDF
  ZNP=DXDA*DYDF-DYDA*DXDF
  RMN=DSQRT(XNP*XNP+YNP*YNP+ZNP*ZNP)
C      WRITE(6,130) DXDA,DZDA,DXDF,DYDF,DZDF,XNP,YNP,ZNP,RMN,TP,AP
C 130 FORMAT(1X,'$$$ ',5F15.7/4X,5F15.7)
  XNP=XNP/RMN
  YNP=YNP/RMN
  ZNP=ZNP/RMN
C      *****
  XNPF=XNP*DCOS(X(7))+YNP*DSIN(X(7))
  YNPF=YNP*DCOS(X(7))-XNP*DSIN(X(7))
  ZNPF=ZNP
  CF2FG=DCOS(X(1))
  SF2FG=DSIN(X(1))
  CF2FGK=DCOS(X(1)+KSIC)
  SF2FGK=DSIN(X(1)+KSIC)
  AG=-X(5)*SB+RG*X(2)
  RFG1=CK*CK/CKC*CF2FGK*AG+RG*SF2FG
  RFG2=CK*CK/CKC*SF2FGK*AG-RG*CF2FG
  RFG3=X(5)*CB-AG*SK*SK*TB
  RNFG1=CK*CB*CF2FG-SK*SF2FG
  RNFG2=CK*CB*SF2FG+SK*CF2FG
  RNFG3=CK*SB

```

```

XGF=RFG1*S(1,1)+RFG2*S(1,2)+RFG3*S(1,3)
YGF=RFG1*S(2,1)+RFG2*S(2,2)+RFG3*S(2,3)
ZGF=RFG1*S(3,1)+RFG2*S(3,2)+RFG3*S(3,3)
XNGF=RNFG1*S(1,1)+RNFG2*S(1,2)+RNFG3*S(1,3)
YNGF=RNFG1*S(2,1)+RNFG2*S(2,2)+RNFG3*S(2,3)
ZNGF=RNFG1*S(3,1)+RNFG2*S(3,2)+RNFG3*S(3,3)
C   WRITE (6,100) X(1),X(2),X(3),X(4),X(5)
C   WRITE (6,100) XPF,YPF,ZPF,XGF,YGF,ZGF
C 100 FORMAT (1X,'%%%%',8E15.7)
C   WRITE (6,100) XNPF,YNPF,ZNPF,XNGF,YNGF,ZNGF
Y(1)=XPF-XGF
Y(2)=YPF-YGF-C
Y(3)=ZPF-ZGF
Y(5)=XNPF-XNGF
Y(4)=ZNPF-ZNGF
X(10)=DSQRT(XPF*XPF+YPF*YPF)
X(11)=DSQRT(XGF*XGF+YGF*YGF)
X(12)=ZPF
X(13)=ZGF
C   WRITE (6,20) (Y(II),II=1,5)
C 20  FORMAT (1X,'$$$$',6F15.7)
      RETURN
      END
C
      SUBROUTINE INTMAT (A,N,M)
C   THIS SUBROUTINE IS USED TO INITIATE THE MATRIX, WITH UNIT DIAGONAL
C   ELEMENTS AND NULL OTHER ELEMENTS
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4)
      DO 5 I=1,N
      DO 5 J=1,M
      A(I,J)=0.
      IF (I.EQ.J) A(I,J)=1.
5  CONTINUE
      RETURN
      END
C
C
      SUBROUTINE NONLIN
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /BLOCK1/ X(13),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
+                EPSI,DELTA,NC,NE,NDIM
C
      DO 5 I=1,NC
      CALL FUNC
C   WRITE (6,10) I,(X(J),Y(J),J=1,5)
C 10  FORMAT(1X,'***',15/5(1X,2D15.7/))
      DO 15 J=1,NE
      IF (DABS(Y(J)).GT.EPSI) GO TO 25
15  CONTINUE
      GO TO 105
25  DO 35 J=1,NE
35  Y1(J)=Y(J)

```

```

DO 45 J=1,NE
DIFF=DABS(X(J))*DELTA
IF (DABS(X(J)).LT.1.D-12) DIFF=DELTA
XMAM=X(J)
X(J)=X(J)-DIFF
CALL FUNC
X(J)=XMAM
DO 55 K=1,NE
55 A(K,J)=(Y1(K)-Y(K))/DIFF
45 CONTINUE
DO 65 J=1,NE
65 Y(J)=-Y1(J)
C DO 205 K=1,NE
C 205 WRITE (6,245) (A(K,J),J=1,NE)
C 245 FORMAT (1X,'---',5D15.7)
CALL DECOMP (NDIM,NE,A,COND,IPVT,WORK)
CALL SOLVE (NDIM,NE,A,Y,IPVT)
DO 75 J=1,NE
75 X(J)=X(J)+Y(J)
5 CONTINUE
C 105 WRITE (6,20) I
C 20 FORMAT (1X,'I=',I2)
105 RETURN
END

C
C
C
SUBROUTINE DECOMP (NDIM,N,A,COND,IPVT,WORK)
C
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(NDIM,N),WORK(N),IPVT(N)
C
C DECOMPOSES AREAL MATRIX BY GAUSSIAN ELIMINATION,
C AND ESTIMATES THE CONDITION OF THE MATRIX.
C
C -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
C M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
C
C USE SUBROUTINE SOLVE TO COMPUTE SOLUTIONS TO LINEAR SYSTEM.
C
C INPUT..
C
C NDIM = DECLARED ROW DIMENSION OF THE ARRAY CONTAINING A
C N = ORDER OF THE MATRIX
C A = MATRIX TO BE TRIANGULARIZED
C
C OUTPUT..
C
C A CONTAINS AN UPPER TRIANGULAR MATRIX U AND A PREMUTED
C VERSION OF A LOWER TRIANGULAR MATRIX I-L SO THAT
C (PERMUTATION MATRIX)*A=L*U
C
C COND = AN ESTIMATE OF THE CONDITION OF A.
C FOR THE LINEAR SYSTEM A*X = B, CHANGES IN A AND B

```

```

C      MAY CAUSE CHANGES COND TIMES AS LARGE IN X.
C      IF COND+1.0 .EQ. COND , A IS SINGULAR TO WORKING
C      PRECISION. COND IS SET TO 1.0D+32 IF EXACT
C      SINGULARITY IS DETECTED.
C
C      IPVT      = THE PIVOT VECTOR
C      IPVT(K)   = THE INDEX OF THE K-TH PIVOT ROW
C      IPVT(N)   = (-1)**(NUMBER OF INTERCHANGES)
C
C      WORK SPACE.. THE VECTOR WORK MUST BE DECLARED AND INCLUDED
C      IN THE CALL. ITS INPUT CONTENTS ARE IGNORED.
C      ITS OUTPUT CONTENTS ARE USUALLY UNIMPORTANT.
C
C      THE DETERMINANT OF A CAN BE OBTAINED ON OUTPUT BY
C      DET(A) = IPVT(N) * A(1,1) * A(2,2) * ... * A(N,N) .
C
C      IPVT(N)=1
C      IF (N.EQ.1) GO TO 150
C      NM1=N-1
C
C      COMPUTE THE 1-NORM OF A .
C      ANORM=0.D0
C      DO 20 J=1,N
C          T=0.D0
C          DO 10 I=1,N
10      T=T+DABS(A(I,J))
C          IF (T.GT.ANORM) ANORM=T
C      20 CONTINUE
C
C      DO GAUSSIAN ELIMINATION WITH PARTIAL
C      PIVOTING.
C      DO 70 K=1,NM1
C          KP1=K+1
C
C      FIND THE PIVOT.
C      M=K
C      DO 30 I=KP1,N
C          IF (DABS(A(I,K)).GT.DABS(A(M,K))) M=I
30      CONTINUE
C      IPVT(K)=M
C      IF (M.NE.K) IPVT(N)=-IPVT(N)
C      T=A(M,K)
C      A(M,K)=A(K,K)
C      A(K,K)=T
C
C      SKIP THE ELIMINATION STEP IF PIVOT IS ZERO.
C      IF (T.EQ.0.D0) GO TO 70
C
C      COMPUTE THE MULTIPLIERS.
C      DO 40 I=KP1,N
40      A(I,K)=-A(I,K)/T
C
C      INTERCHANGE AND ELIMINATE BY COLUMNS.
C      DO 60 J=KP1,N
C          T=A(M,J)
C          A(M,J)=A(K,J)
C          A(K,J)=T
C          IF (T.EQ.0.D0) GO TO 60
C      DO 50 I=KP1,N

```

```

50     A(I,J)=A(I,J)+A(I,K)*T
60     CONTINUE
70     CONTINUE
C
C COND = (1-NORM OF A)*(AN ESTIMATE OF THE 1-NORM OF A-INVERSE)
C THE ESTIMATE IS OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
C SMALL SINGULAR VECTOR. THIS INVOLVES SOLVING TWO SYSTEMS
C OF EQUATIONS, (A-TRANPOSE)*Y = E AND A*Z = Y WHERE E
C IS A VECTOR OF +1 OR -1 COMPONENTS CHOSEN TO CAUSS GROWTH IN Y.
C ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)
C
C                                     SOLVE (A-TRANPOSE)*Y = E .
      DO 100 K=1,N
        T=0.DO
        IF (K.EQ.1) GO TO 90
        KM1=K-1
        DO 80 I=1,KM1
90      T=T+A(I,K)*WORK(I)
        EK=1.DO
        IF (T.LT.0.DO) EK=-1.DO
        IF (A(K,K).EQ.0.DO) GO TO 160
100    WORK(K)=- (EK+T)/A(K,K)
        DO 120 KB=1,NM1
          K=N-KB
          T=0.DO
          KP1=K+1
          DO 110 I=KP1,N
110      T=T+A(I,K)*WORK(K)
          WORK(K)=T
          M=IPVT(K)
          IF (M.EQ.K) GO TO 120
          T=WORK(M)
          WORK(M)=WORK(K)
          WORK(K)=T
120    CONTINUE
C
      YNORM=0.DO
      DO 130 I=1,N
130    YNORM=YNORM+DABS(WORK(I))
C
C                                     SOLVE A*Z = Y
      CALL SOLVE (NDIM,N,A,WORK,IPVT)
C
      ZNORM=0.DO
      DO 140 I=1,N
140    ZNORM=ZNORM+DABS(WORK(I))
C
C                                     ESTIMATE THE CONDITION.
      COND=ANORM*ZNORM/YNORM
      IF (COND.LT.1.DO) COND=1.DO
      RETURN
C
C                                     1-BY-1 CASE..
150   COND=1.DO
      IF (A(1,1).NE.0.DO) RETURN

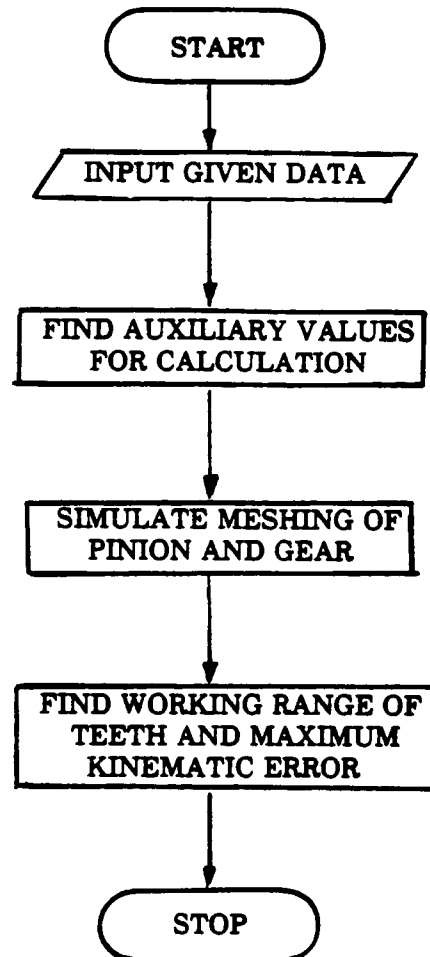
```

```

C
C                                     EXACT SINGULARITY
160 COND=1.0D32
    RETURN
    END
    SUBROUTINE SOLVE (NDIM,N,A,B,IPVT)
C
    IMPLICIT REAL*8 (A-H,O-Z)
    DIMENSION A(NDIM,N),B(N),IPVT(N)
C
C  SOLVES A LINEAR SYSTEM,  A*X = B
C  DO NOT SOLVE THE SYSTEM IF DECOMP HAS DETECTED SINGULARITY.
C
C  -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
C    M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
C
C  INPUT..
C
C    NDIM = DECLARED ROW DIMENSION OF ARRAY CONTAINING  A
C    N     = ORDER OF MATRIX
C    A     = TRIANGULARIZED MATRIX OBTAINED FROM SUBROUTINE DECOMP
C    B     = RIGHT HAND SIDE VECTOR
C    IPVT  = PIVOT VECTOR OBTAINED FROM DECOMP
C
C  OUTPUT..
C
C    B = SOLUTION VECTOR,  X
C
C                                     DO THE FORWARD ELIMINATION.
    IF (N.EQ.1) GO TO 50
    NM1=N-1
    DO 20 K=1,NM1
        KP1=K+1
        M=IPVT(K)
        T=B(M)
        B(M)=B(K)
        B(K)=T
        DO 10 I=KP1,N
10      B(I)=B(I)+A(I,K)*T
20    CONTINUE
C
C                                     NOW DO THE BACK SUBSTITUTION.
    DO 40 KB=1,NM1
        KM1=N-KB
        K=KM1+1
        B(K)=B(K)/A(K,K)
        T=-B(K)
        DO 30 I=1,KM1
30      B(I)=B(I)+A(I,K)*T
40    CONTINUE
50  B(1)=B(1)/A(1,1)
    RETURN
    END
C
C

```

# FLOWCHART FOR PROGRAM VI



```

C... *****
C... *
C... *
C... *
C... *          PROGRAM VI
C... * TRANSMISSION ERRORS OF HELICAL GEARS WITH THEIR
C... *          AXES DEFORMED BY INTERACTING FORCE
C... *
C... *          AUTHORS: FAYDOR LITVIN
C... *                   JIAO ZHANG
C... *
C... *****
C
C PURPOSE
C
C THIS PROGRAM IS USED TO CALCULATE THE TRANSMISSION ERRORS OF A
C HELICAL PINION AND A HELICAL GEAR IN MESHING WHEN THEIR AXES ARE
C DEFORMED BY INTERACTING FORCE (BOTH PINION AND GEAR ARE NOT CROWNED)
C
C NOTE
C
C THIS PROGRAM IS WRITTEN IN FORTRAN 77. IT CAN BE COMPILED BY V
C COMPILER IN IBM MAINFRAME OR FORTRAN COMPILER IN VAX SYSTEM.
C
C IMPLICIT REAL*8(A-H,O-Z)
C DOUBLE PRECISION KSIN,MU,MUO,KSIC
C DIMENSION Z(99),ANGLE(99),ERROR(99),ERR(99),W(99),RPR(99),RGR(99),
C + S1(4,4),S2(4,4),ZP(99)
C COMMON /BLOCK1/ X(12),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
C + EPSI,DELTA,NC,NE,NDIM,NCTL,CX2
C COMMON /BLOCK2/ S(4,4),C,RP,RG,CK,SK,CB,SB,CKC,KSIC,ZG,TB,ZNPF,
C + COEG1,COEG2,CB1,SB1,TB1
C
C DEFINE PARAMETERS USED BY PROGRAMS
C
C (1) IN ANG LP ARE UNIT NUMBERS ASSIGNED TO THE INPUT AND OUTPUT
C DEVICES
C IN=5
C LP=6
C (2) NDBUG IS USE TO CONTROL THE AUXILIARY OUTPUT FOR DEBUGGING
C NDBUG=1
C (3) NC IS THE UPPER LIMITATION OF REPEATATION FOR SOLVING NONLINEAR
C EQUATIONS;
C EPSI IS THE CLEARANCE OF FUNCTION VALUES WHEN THE FUNCTIONS
C IS CONSIDERED AS SOLVED (ALL FUNCTIONS HAVE FORMS OF F(X)=0);
C DELTA IS THE RELATIVE DIFFERENCE FOR TAKING DERIVATIVES
C NC, EPSI AND DELTA MAY BE CHANGED WHEN SOLUTIONS ARE DIVERGENT
C OR LESS ACCURATE
C NC=100
C DELTA=1.D-4
C EPSI=1.D-13
C (4) OTHER PARAMETERS (DON'T CHANGE)
C NDIM=10
C DR=DATAN(1.D0)/45.D0

```

```

C  DEFINE INPUT PARAMTERS OF PROBLEM(USE INCH AS UNIT OF LENGTH)
C  (1) PINION AND GEAR: PN=DIAMETRAL PITCH; NP=PINION TOOTH NUMBER;
C      RMPG=TOOTH NUMBER RATIO(GEAR TOOTH NO./NP);
C      KSIN=PRESSURE ANGLE IN NORMAL SECTION;
C      BETAP=HELIX ANGLE OF PINION AND GEAR;
C      PAD=HEIGHT OF ADDENDUM OF PINION;
C      GAD=HEIGHT OF ADDENDUM OF GEAR;
C      ZG=GEAR TOOTH LENGTH (PINION TOOTH LENGTH IS LONGER)
C      COE=COEFF. OF CENTRAL DISTANCE(USUALLY COE=1.)
C      PN=10.D0
C      NP=20
C      RMPG=2.D0
C      KSIN=20.D0*DR
C      BETAP=15.D0*DR
C      PAD=1.0/PN
C      GAD=1.0/PN
C      ZG= 5./PN
C      COE=1.000D0
C  (2) SHAFT DEFORMATION:
C      NSIM=MODEL ID NO.(1=SIMPLIFIED DEFORMATION MATRIX;
C                          2=UNSIMPLIFIED DEFORMATION MATRIX)
C      RLAMP=PINION SHAFT ROTATION
C      RLAGM=GEAR SHAFT ROTATION
C      RVP=PINION SHAFT DEFLECTION
C      RVG=GEAR SHAFT DEFLECTION
C
C      NSIM=1
C      RLAMP= 2./60.*DR
C      RLAGM= 2./60.*DR
C      RVP=0.0125
C      RVG=0.0125
C  (3) OUTPUT: FEEI=INCREMENT OF ROTATION ANGLE OF PINION(DEGREEE)
C      FEEI=1.0D0*DR
C
C  DESCRIPTION OF OUTPUT PARAMERTERS
C
C      FEE1=ROTATION ANGLE OF PINION
C      FEE2=ROTATION ANGLE OF GEAR
C      RP=RADIUS OF PINION CONTACT POINT
C      RG=RADIUS OF GEAR CONTACT POINT
C
C  FIND AUXILIARY VALUES FOR CALCULATION
C
C  DEFINE USEFUL CONSTANTS AND PARAMETERS FOR PINION AND GEAR
C      DELTAB=DR/60.*( 0.D0)
C      BETAP1=BETAP+DELTAB
C      CLP=DCOS(RLAMP)
C      SLP=DSIN(RLAMP)
C      CLG=DCOS(RLAGM)
C      SLG=DSIN(RLAGM)
C      WRITE (LP,4) CLP,SLP,CLG,SLG
C  4  FORMAT(1X,4F15.7)
C      SK=DSIN(KSIN)

```

```

CK=DCOS(KSIN)
SB=DSIN(BETAP)
CB=DCOS(BETAP)
TB=SB/CB
SB1=DSIN(BETAP1)
CB1=DCOS(BETAP1)
TB1=SB1/CB1
KSIC=DATAN(SK/CK/CB)
CKC=DCOS(KSIC)
SKC=DSIN(KSIC)
PT=PN*CB
RP=NP/2./PT
RPA=RP+PAD
RPATOL=RPA-0.0005D0
RG=RP*RMPG
RGA=RG+GAD
RGATOL=RGA-0.0005D0
WRITE (LP,56) RPATOL, RGATOL
56 FORMAT (1X, '#####', 'RPA=', F15.7, 5X, 'RGA=', F15.7)
COEG1=1.+SK*SK*TB*TB
COEG2=RG/COEG1
ZNPf=CK*SB1
C=(RP+RG)*COE
CALL INTMAT(S,4,4)
CALL INTMAT(S1,4,4)
CALL INTMAT(S2,4,4)
DO 505 LL=1,2
NSIM=LL
IF (NSIM.EQ.1) THEN
WRITE (LP,500)
500 FORMAT (1H1,///,1X,
+ 'THE CASE OF SIMPLIFIED DEFORMATION MATRIX OF GEAR AXES')
S(1,3)=(RLAMP+RLAMG)*CKC
S(2,3)=(RLAMP+RLAMG)*SKC
S(3,1)=-S(1,3)
S(3,2)=-S(2,3)
S(1,4)=(RVP+RVG)*CKC
S(2,4)=(RVP+RVG)*SKC+C
S(3,4)=-C*RLAMP*SKC
ELSE
WRITE (LP,501)
501 FORMAT (1H1,///,1X,
+ 'THE CASE OF UNSIMPLIFIED DEFORMATION MATRIX OF GEAR AXES')
S1(1,1)=1.+CKC*CKC*(CLP-1.)
S1(1,2)=SKC*CKC*(CLP-1.)
S1(1,3)=CKC*SLP
S1(2,1)=S1(1,2)
S1(2,2)=1.+SKC*SKC*(CLP-1.)
S1(2,3)=SKC*SLP
S1(3,1)=-S1(1,3)
S1(3,2)=-S1(2,3)
S1(3,3)=CLP
S1(1,4)=RVP*CKC*CLP
S1(2,4)=RVP*SKC*CLP

```

```

S1(3,4)=-RVP*SLP
S2(1,1)=1.+CKC*CKC*(CLG-1.)
S2(1,2)=SKC*CKC*(CLG-1.)
S2(1,3)=CKC*SLG
S2(2,1)=S1(1,2)
S2(2,2)=1.+SKC*SKC*(CLG-1.)
S2(2,3)=SKC*SLG
S2(3,1)=-S1(1,3)
S2(3,2)=-S1(2,3)
S2(3,3)=CLG
S2(1,4)=RVG*CKC
S2(2,4)=RVG*SKC+C
S2(3,4)=0.
DO 12 MMI=1,3
DO 12 MMJ=1,4
S(MMI,MMJ)=0.
DO 12 MMK=1,4
S(MMI,MMJ)=S(MMI,MMJ)+S1(MMI,MMK)*S2(MMK,MMJ)
12 CONTINUE
END IF
NCOEF=IDINT(360.*DR/FEEI/FLOAT(NP)+0.5)
N=3*NCOEF
NHALF=(N+1)/2
XIN=0.
DO 205 L=1,2
LSGN=(-1)**L
NCTL=0
NE=4
DO 5 I=1,NE
5 X(I)=0.DO
X(10)=0.
X(11)=0.
DO 15 I=1,NHALF
IF (NCTL.EQ.2) GOTO 205
LI=NHALF+LSGN*(I-1)
IF (L.EQ.1) NMIN=LI
IF (L.EQ.2) NMAX=LI
X(7)=LSGN*FEEI*FLOAT(I-1)
IF (X(10).LE.RPATOL) THEN
X(4)=-X(7)
ELSE
NE=3
NCTL=NCTL+1
END IF
IF (X(11).LT.RGATOL) THEN
X(2)=X(7)/RMPG
ELSE
NCTL=NCTL+1
CX2=X(2)
END IF
X(3)=(ZG-RP*X(4)*SK*SK*TB)/CB/COEG1
CALL NONLIN
X(8)=X(1)+X(2)
C FIND INITIAL VALUE OF X(8)

```

```

      IF (L.EQ.1.AND.I.EQ.1) THEN
      XIN=X(8)
      GO TO 15
      END IF
C
      X(8)=X(8)-XIN
      W(LI)=X(7)/DR
      Z(LI)=X(8)/DR
      ERR(LI)=(X(8)-X(7)/RMPG)*3600.DO/DR
      RPR(LI)=X(10)
      RGR(LI)=X(11)
      ZP(LI)=X(12)
C      WRITE (LP,21) LI,W(LI),Z(LI),ERR(LI),RPR(LI),RGR(LI)
C 21  FORMAT (1X,I2,1X,5F15.7,F15.7)
      15  CONTINUE
205  CONTINUE
      WRITE (LP,10)
10  FORMAT (///8X,'FEE1(D)',8X,'FEE2(D)',8X,'K-ERROR(S)',5X,
+          'RP',13X,'RG',F15.7/)
      DO 25 I=NMIN,NMAX
      WRITE (LP,20) W(I),Z(I),ERR(I),RPR(I),RGR(I)
20  FORMAT (1X,5F15.7,F15.7)
25  CONTINUE
      NT=NMAX-NCOEF
      FII=FEEI/DR/2.DO*FLOAT(NCOEF)
      WRITE (LP,80)
80  FORMAT (//,' FIND THE WORKING RANGE FOR ONE TOOTH:',F15.7/)
      DO 55 I=NMIN,NT
      X(7)=FEEI*FLOAT(I-(N+1)/2)/DR/2.DO
      X(8)=X(7)+FII
      KK=I+NCOEF
      ANGLE(I)=Z(KK)-Z(I)
      ERROR(I)=(ANGLE(I)-FII)*3600.DO
      WRITE (LP,60) X(7),X(8),ANGLE(I),ERROR(I)
60  FORMAT (1X,(' ',F7.2,'----',F7.2,'): ',F15.7,F15.7)
55  CONTINUE
      DO 95 I=NMIN,NT
      ATEMP2=ERROR(I)
      IF (I.NE.NMIN) THEN
      IF (ATEMP1*ATEMP2.LE.0.DO) GOTO 105
      END IF
      ATEMP1=ATEMP2
95  CONTINUE
      WRITE (LP,160)
160  FORMAT (//1X,'MESHING IS DISCONTINUOUS')
      GO TO 505
105  IF (DABS(ATEMP1).LT.DABS(ATEMP2)) I=I-1
      EMAX=0.
      EMIN=0.
      NTEMP=NCOEF+1
      DO 135 J=1,NTEMP
      KS=I+J-1
      ET=ERR(KS)
      IF (ET.LT.EMIN) EMIN=ET

```

```

      IF (ET.GT.EMAX) EMAX=ET
135  CONTINUE
      ET=EMAX-EMIN
      KK=I+NCOEF
      WRITE (LP,170) Z(I),Z(KK),ET
170  FORMAT (//1X,'WORKING RANGE FOR ONE TOOTH: ',F7.2,'----',F7.2/
+          1X,'THE MAXIMUM KINEMATIC ERROR: ',F15.7,' (S)',I2)
505  CONTINUE
      STOP
      END

C
      SUBROUTINE FUNC
C
      IMPLICIT REAL*8(A-H,O-Z)
      DOUBLE PRECISION KSIN,MU,MUO,KSIC
      COMMON /BLOCK1/ X(12),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
+          EPSI,DELTA,NC,NE,NDIM,NCTL,CX2
      COMMON /BLOCK2/ S(4,4),C,RP,RG,CK,SK,CB,SB,CKC,KSIC,ZG,TB,ZNPF,
+          COEG1,COEG2,CB1,SB1,TB1
C  X(4)=FEED; X(7)=FEE1
      X(6)=X(4)+X(7)
      CF1FP=DCOS(X(6))
      SF1FP=DSIN(X(6))
      AP=X(3)*SB1+RP*X(4)
      XPF=-CK*CK/CKC*DCOS(X(6)-KSIC)*AP+RP*SF1FP
      YPF=CK*CK/CKC*DSIN(X(6)-KSIC)*AP+RP*CF1FP
      ZPF=X(3)*CB1+AP*SK*SK*TB1
      XNPF=CK*CB1*CF1FP+SK*SF1FP
      YNPF=-CK*CB1*SF1FP+SK*CF1FP
C  ZNPF=CK*SB1
      CF2FG=DCOS(X(1))
      SF2FG=DSIN(X(1))
      CF2FGK=DCOS(X(1)+KSIC)
      SF2FGK=DSIN(X(1)+KSIC)
      AG1=(-ZG*TB+RG*X(2))/COEG1
      RFG1=CK*CK/CKC*CF2FGK*AG1+RG*SF2FG
      RFG2=CK*CK/CKC*SF2FGK*AG1-RG*CF2FG
C  RFG3=ZG
      RTFG1=CK*CK/CKC*(AG1*SF2FGK+COEG2*CF2FGK)-RG*CF2FG
      RTFG2=CK*CK/CKC*(-AG1*CF2FGK+COEG2*SF2FGK)-RG*SF2FG
C  RTFG3=0.
      XGF=RFG1*S(1,1)+RFG2*S(1,2)+ZG*S(1,3)+S(1,4)
      YGF=RFG1*S(2,1)+RFG2*S(2,2)+ZG*S(2,3)+S(2,4)
      ZGF=RFG1*S(3,1)+RFG2*S(3,2)+ZG*S(3,3)+S(3,4)
      XTGF=RTFG1*S(1,1)+RTFG2*S(1,2)
      YTGF=RTFG1*S(2,1)+RTFG2*S(2,2)
      ZTGF=RTFG1*S(3,1)+RTFG2*S(3,2)
C  WRITE (6,100) RTFG1,RTFG2,AG1,COEG2,COEG1,CF2FGK,SF2FGK
C  WRITE (6,100) X(1),X(2),X(3),X(4),X(5)
C  WRITE (6,100) XPF,YPF,ZPF,XGF,YGF,ZGF
C 100  FORMAT (1X,'%%%%%%%%',8E15.7)
C  WRITE (6,100) XNPF,YNPF,ZNPF,XTGF,YTGF,ZTGF
      Y(1)=XPF-XGF
      Y(2)=YPF-YGF

```

```

      Y(3)=ZPF-ZGF
      IF (NCTL.NE.0) THEN
      Y(4)=X(2)-CX2
      ELSE
      Y(4)=XNPF*XTGF+YNPF*YTGF+ZNPF*ZTGF
      END IF
      X(10)=DSQRT(XPF*XPF+YPF*YPF)
      X(11)=DSQRT(RFG1*RFG1+RFG2*RFG2)
      X(12)=ZPF
C      WRITE (6,20) (Y(II),II=1,4)
C 20  FORMAT (1X,'$$$$',6F15.7)
      RETURN
      END

C
      SUBROUTINE INTMAT (A,N,M)
C  THIS SUBROUTINE IS USED TO INITIATE THE MATRIX, WITH UNIT DIAGONAL
C  ELEMENTS AND NULL OTHER ELEMENTS
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4)
      DO 5 I=1,N
      DO 5 J=1,M
      A(I,J)=0.
      IF (I.EQ.J) A(I,J)=1.
5  CONTINUE
      RETURN
      END

C
C
      SUBROUTINE NONLIN
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /BLOCK1/ X(12),Y(10),A(10,10),Y1(10),IPVT(10),WORK(10),
+          EPSI,DELTA,NC,NE,NDIM,NCTL,CX2
C
      DO 5 I=1,NC
      CALL FUNC
C      WRITE (6,10) I,(X(J),Y(J),J=1,4)
C 10  FORMAT(1X,'***',I5/5(1X,2D15.7/))
      DO 15 J=1,NE
      IF (DABS(Y(J)).GT.EPSI) GO TO 25
15  CONTINUE
      GO TO 105
25  DO 35 J=1,NE
35  Y1(J)=Y(J)
      DO 45 J=1,NE
      DIFF=DABS(X(J))*DELTA
      IF (DABS(X(J)).LT.1.D-12) DIFF=DELTA
      XMAM=X(J)
      X(J)=X(J)-DIFF
      CALL FUNC
      X(J)=XMAM
      DO 55 K=1,NE
55  A(K,J)=(Y1(K)-Y(K))/DIFF
45  CONTINUE

```

```

      DO 65 J=1,NE
65  Y(J)=-Y1(J)
C    DO 85 K=1,NE
C 85  WRITE (6,104) (A(K,J),J=1,NE),Y(K)
C 104 FORMAT(1X,'A',2X,5E15.7)
      CALL DECOMP (NDIM,NE,A,COND,IPVT,WORK)
      CALL SOLVE (NDIM,NE,A,Y,IPVT)
      DO 75 J=1,NE
75  X(J)=X(J)+Y(J)
      5 CONTINUE
105  RETURN
      END

C
C
C
      SUBROUTINE DECOMP (NDIM,N,A,COND,IPVT,WORK)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NDIM,N),WORK(N),IPVT(N)
C
C  DECOMPOSES AREAL MATRIX BY GAUSSIAN ELIMINATION,
C  AND ESTIMATES THE CONDITION OF THE MATRIX.
C
C  -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
C    M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
C
C  USE SUBROUTINE SOLVE TO COMPUTE SOLUTIONS TO LINEAR SYSTEM.
C
C  INPUT..
C
C    NDIM = DECLARED ROW DIMENSION OF THE ARRAY CONTAINING  A
C    N    = ORDER OF THE MATRIX
C    A    = MATRIX TO BE TRIANGULARIZED
C
C  OUTPUT..
C
C    A  CONTAINS AN UPPER TRIANGULAR MATRIX U  AND A PREMUTED
C    VERSION OF A LOWER TRIANGULAR MATRIX I-L  SO THAT
C    (PERMUTATION MATRIX)*A=L*U
C
C    COND = AN ESTIMATE OF THE CONDITION OF  A.
C    FOR THE LINEAR SYSTEM  $A * X = B$ , CHANGES IN  A  AND  B
C    MAY CAUSE CHANGES COND TIMES AS LARGE IN  X.
C    IF  $COND + 1.0 \approx COND$ , A IS SINGULAR TO WORKING
C    PRECISION. COND IS SET TO 1.0D+32 IF EXACT
C    SINGULARITY IS DETECTED.
C
C    IPVT      = THE PIVOT VECTOR
C    IPVT(K)   = THE INDEX OF THE K-TH PIVOT ROW
C    IPVT(N)   = (-1)**(NUMBER OF INTERCHANGES)
C
C  WORK SPACE.. THE VECTOR WORK MUST BE DECLARED AND INCLUDED
C    IN THE CALL. ITS INPUT CONTENTS ARE IGNORED.
C    ITS OUTPUT CONTENTS ARE USUALLY UNIMPORTANT.

```

```

C
C THE DETERMINANT OF A CAN BE OBTAINED ON OUTPUT BY
C   DET(A) = IPVT(N) * A(1,1) * A(2,2) * ... * A(N,N) .
C
C   IPVT(N)=1
C   IF (N.EQ.1) GO TO 150
C   NM1=N-1
C
C           COMPUTE THE 1-NORM OF A .
C
C   ANORM=0.D0
C   DO 20 J=1,N
C       T=0.D0
C       DO 10 I=1,N
10      T=T+DABS(A(I,J))
C       IF (T.GT.ANORM) ANORM=T
20  CONTINUE
C
C           DO GAUSSIAN ELIMINATION WITH PARTIAL
C           PIVOTING.
C
C   DO 70 K=1,NM1
C       KP1=K+1
C
C           FIND THE PIVOT.
C
C       M=K
C       DO 30 I=KP1,N
C           IF (DABS(A(I,K)).GT.DABS(A(M,K))) M=I
30  CONTINUE
C       IPVT(K)=M
C       IF (M.NE.K) IPVT(N)=-IPVT(N)
C       T=A(M,K)
C       A(M,K)=A(K,K)
C       A(K,K)=T
C
C           SKIP THE ELIMINATION STEP IF PIVOT IS ZERO.
C       IF (T.EQ.0.D0) GO TO 70
C
C           COMPUTE THE MULTIPLIERS.
C
C       DO 40 I=KP1,N
40  A(I,K)=-A(I,K)/T
C
C           INTERCHANGE AND ELIMINATE BY COLUMNS.
C
C       DO 60 J=KP1,N
C           T=A(M,J)
C           A(M,J)=A(K,J)
C           A(K,J)=T
C           IF (T.EQ.0.D0) GO TO 60
C           DO 50 I=KP1,N
50      A(I,J)=A(I,J)+A(I,K)*T
60  CONTINUE
70  CONTINUE
C
C COND = (1-NORM OF A)*(AN ESTIMATE OF THE 1-NORM OF A-INVERSE)
C THE ESTIMATE IS OBTAINED BY ONE STEP OF INVERSE ITERATION FOR THE
C SMALL SINGULAR VECTOR. THIS INVOLVES SOLVING TWO SYSTEMS
C OF EQUATIONS, (A-TRANPOSE)*Y = E AND A*Z = Y WHERE E
C IS A VECTOR OF +1 OR -1 COMPONENTS CHOSEN TO CAUSS GROWTH IN Y.
C ESTIMATE = (1-NORM OF Z)/(1-NORM OF Y)
C
C           SOLVE (A-TRANPOSE)*Y = E .

```

```

      DO 100 K=1,N
        T=0.D0
        IF (K.EQ.1) GO TO 90
        KM1=K-1
        DO 80 I=1,KM1
80      T=T+A(I,K)*WORK(I)
90      EK=1.D0
        IF (T.LT.0.D0) EK=-1.D0
        IF (A(K,K).EQ.0.D0) GO TO 160
100     WORK(K)=- (EK+T)/A(K,K)
        DO 120 KB=1,NM1
          K=N-KB
          T=0.D0
          KP1=K+1
          DO 110 I=KP1,N
110      T=T+A(I,K)*WORK(K)
          WORK(K)=T
          M=IPVT(K)
          IF (M.EQ.K) GO TO 120
          T=WORK(M)
          WORK(M)=WORK(K)
          WORK(K)=T
120     CONTINUE
C
      YNORM=0.D0
      DO 130 I=1,N
130     YNORM=YNORM+DABS(WORK(I))
C
C                                     SOLVE  A*Z = Y
      CALL SOLVE (NDIM,N,A,WORK,IPVT)
C
      ZNORM=0.D0
      DO 140 I=1,N
140     ZNORM=ZNORM+DABS(WORK(I))
C
C                                     ESTIMATE THE CONDITION.
      COND=ANORM*ZNORM/YNORM
      IF (COND.LT.1.D0) COND=1.D0
      RETURN
C
C                                     1-BY-1 CASE..
150     COND=1.D0
      IF (A(1,1).NE.0.D0) RETURN
C
C                                     EXACT SINGULARITY
160     COND=1.0D32
      RETURN
      END
      SUBROUTINE SOLVE (NDIM,N,A,B,IPVT)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NDIM,N),B(N),IPVT(N)
C
C   SOLVES A LINEAR SYSTEM,  A*X = B
C   DO NOT SOLVE THE SYSTEM IF DECOMP HAS DETECTED SINGULARITY.

```

```

C
C -COMPUTER METHODS FOR MATHEMATICAL COMPUTATIONS-, BY G. E. FORSYTHE,
C   M. A. MALCOLM, AND C. B. MOLER (PRENTICE-HALL, 1977)
C
C INPUT..
C
C   NDIM = DECLARED ROW DIMENSION OF ARRAY CONTAINING  A
C   N    = ORDER OF MATRIX
C   A    = TRIANGULARIZED MATRIX OBTAINED FROM SUBROUTINE DECOMP
C   B    = RIGHT HAND SIDE VECTOR
C   IPVT = PIVOT VECTOR OBTAINED FROM DECOMP
C
C OUTPUT..
C
C   B = SOLUTION VECTOR,  X
C
C                                     DO THE FORWARD ELIMINATION.
C   IF (N.EQ.1) GO TO 50
C   NM1=N-1
C   DO 20 K=1,NM1
C     KP1=K+1
C     M=IPVT(K)
C     T=B(M)
C     B(M)=B(K)
C     B(K)=T
C     DO 10 I=KP1,N
10    B(I)=B(I)+A(I,K)*T
20  CONTINUE
C
C                                     NOW DO THE BACK SUBSTITUTION.
C   DO 40 KB=1,NM1
C     KM1=N-KB
C     K=KM1+1
C     B(K)=B(K)/A(K,K)
C     T=-B(K)
C     DO 30 I=1,KM1
30    B(I)=B(I)+A(I,K)*T
40  CONTINUE
50  B(1)=B(1)/A(1,1)
C   RETURN
C   END
C
C

```

# Report Documentation Page

1. Report No. <b>NASA CR-4224 AVSCOM TR 89-C-002</b>		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle  <b>Topology of Modified Helical Gears and Tooth Contact Analysis (TCA) Program</b>				5. Report Date  <b>April 1989</b>	
				6. Performing Organization Code	
7. Author(s)  <b>Faydor L. Litvin and Jiao Zhang</b>				8. Performing Organization Report No.  <b>None (E-4626)</b>	
				10. Work Unit No. <b>1L162209A47A 505-63-51</b>	
9. Performing Organization Name and Address  <b>University of Illinois at Chicago Department of Mechanical Engineering Chicago, Illinois</b>				11. Contract or Grant No.  <b>NAG3-655</b>	
				13. Type of Report and Period Covered  <b>Contractor Report Final</b>	
12. Sponsoring Agency Name and Address  <b>Propulsion Directorate U.S. Army Aviation Research and Technology Activity—AVSCOM Cleveland, Ohio 44135-3127 and NASA Lewis Research Center Cleveland, Ohio 44135-3191</b>				14. Sponsoring Agency Code	
15. Supplementary Notes  <b>Project Manager, Robert F. Handschuh, Propulsion Directorate, U.S. Army Aviation Research and Technology Activity—AVSCOM, NASA Lewis Research Center.</b>					
16. Abstract  <b>The contents of this report covers: (i) development of optimal geometry for crowned helical gears; (ii) method for their generation; (iii) tooth contact analysis (TCA) computer programs for the analysis of meshing and bearing contact of the crowned helical gears and (iv) modeling and simulation of gear shaft deflection. The developed method for synthesis is used for determination of optimal geometry for crowned helical pinion surface and is directed to localize the bearing contact and guarantee the favorable shape and low level of the transmission errors. Two new methods for generation of the crowned helical pinion surface have been proposed. One is based on application of the tool with a surface of revolution that slightly deviates from a regular cone surface. The tool can be used as a grinding wheel or as a shaver. The other is based on crowning pinion tooth surface with predesigned transmission errors. The pinion tooth surface can be generated by a computer controlled automatic grinding machine. The TCA program simulates the meshing and bearing contact of misaligned gears. The transmission errors are also determined. The gear shaft deformation has been modeled and investigated. It has been found that the deflection of gear shafts has the same effects as those of gear misalignment.</b>					
17. Key Words (Suggested by Author(s))  <b>Helical gears; Crowned gears; Gear geometry; Tooth contact analysis</b>				18. Distribution Statement  <b>Unclassified—Unlimited Subject Category 37</b>	
19. Security Classif. (of this report)  <b>Unclassified</b>		20. Security Classif. (of this page)  <b>Unclassified</b>		22. Price*  <b>A08</b>	
		21. No of pages  <b>160</b>			